THE UNIVERSITY *of* EDINBURGH

Dynamic Topic Adaptation for Improved Contextual Modelling in Statistical Machine Translation

Eva Hasler

Doctor of Philosophy

Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2014

Abstract

In recent years there has been an increased interest in domain adaptation techniques for statistical machine translation (SMT) to deal with the growing amount of data from different sources. Topic modelling techniques applied to SMT are closely related to the field of domain adaptation but more flexible in dealing with unstructured text. Topic models can capture latent structure in texts and are therefore particularly suitable for modelling structure in between and beyond corpus boundaries, which are often arbitrary.

In this thesis, the main focus is on dynamic translation model adaptation to texts of unknown origin, which is a typical scenario for an online MT engine translating web documents. We introduce a new bilingual topic model for SMT that takes the entire document context into account and for the first time directly estimates topic-dependent phrase translation probabilities in a Bayesian fashion. We demonstrate our model's ability to improve over several domain adaptation baselines and further provide evidence for the advantages of bilingual topic modelling for SMT over the more common monolingual topic modelling. We also show improved performance when deriving further adapted translation features from the same model which measure different aspects of topical relatedness.

We introduce another new topic model for SMT which exploits the distributional nature of phrase pair meaning by modelling topic distributions over phrase pairs using their distributional profiles. Using this model, we explore combinations of local and global contextual information and demonstrate the usefulness of different levels of contextual information, which had not been previously examined for SMT. We also show that combining this model with a topic model trained at the document-level further improves performance. Our dynamic topic adaptation approach performs competitively in comparison with two supervised domain-adapted systems.

Finally, we shed light on the relationship between domain adaptation and topic adaptation and propose to combine multi-domain adaptation and topic adaptation in a framework that entails automatic prediction of domain labels at the document level. We show that while each technique provides complementary benefits to the overall performance, there is an amount of overlap between domain and topic adaptation. This can be exploited to build systems that require less adaptation effort at runtime.

Lay Summary

Automatic translation of written text is an active area of research and the performance of automatic translation systems has improved significantly in recent years. The most common approach is to translate text sentence by sentence, without taking the information contained in previous sentences into account. Within the same sentence, only information from neighbouring regions is typically used. As a result, most translation systems still face problems when translating text that contains ambiguous words which have different meanings in different contexts. Choosing the correct translation of a word in a given context - which may be the surrounding sentence, paragraph or document - requires incorporating a representation of contextual information into the translation system.

The work in this thesis addresses the issue of translating words using information from the context in order to preserve the correct meaning in translations. We propose different ways of incorporating such information into the translation system and compare their performance. An important concept for these context representations is the notion of underlying topics. For example, *politics*, *art* or *holidays* could be labels for underlying topics. However, these labels are not given for a document under translation and therefore the first step for the translation system is to detect which underlying topics are present in a given text. The second step is to choose translations for source words and phrases which are likely translations in the given context.

We propose two new models for enhancing a translation system with contextual information. The first model operates at the document level and effectively uses a specialised translation system for each document under translation. Adapting these different translation systems is an iterative process: knowledge about the underlying topics is refined until the system is confident that the quality of the topic representations is sufficient. The second model can work at the document level or at the sentence level and can further combine the information from both contexts. It differs from the first model in the way that the training data - the data used to automatically learn the model - is structured during the learning phase. While for the first model, information about the underlying topics is associated with text documents, in the second model this information is associated with the translation fragments which are the building blocks of an automatic translation.

Both of the proposed models are shown to improve translation quality in comparison to simpler translation models which lack the ability to use information from the context.

Acknowledgements

First, I would like to thank my supervisor, Philipp Koehn, for his support and advice throughout the years and for opening many doors for me. Many thanks also to my second supervisor, Barry Haddow, for his advice and critical feedback. I would especially like to thank Phil Blunsom for his guidance during the second half of my PhD which has been invaluable for the core part of this thesis. Thanks to Abhishek Arun for a push in the right direction and to Hieu for answering many questions related to Moses. Thanks to Philipp, Barry, Diego, Rico and Giuseppe for their help with proof-reading this thesis as well as to my examiners Mirella Lapata and Holger Schwenk for their time and effort.

It has been a great experience to be part of the machine translation group as well as the NLP group in Edinburgh. Thanks to the many colleagues, friends and flatmates in Edinburgh who have made the past years such an enjoyable and memorable time for me. Finally, I want to thank my parents, my brother and Giuseppe for their support and encouragement.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Eva Hasler*)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automatic machine translation systems have been around for several decades, but since the emergence of data-driven, *statistical machine translation* (SMT) systems (Weaver, 1955; Brown et al., 1990, 1993) the field has started to develop rapidly, in particular with the introduction of phrase-based systems (Och and Ney, 2002; Koehn et al., 2003; Och and Ney, 2004; Koehn, 2004a). The easier access to large computing resources in recent years has lead to the development and application of new algorithms for machine translation. The ongoing efforts to produce high-quality parallel corpora as well as the vastly growing amounts of data on the web mean that more and more natural language data is available for training translation systems. On the one hand, these free data sources are often quite noisy and diverse and raise the question as to how this data is best used for training translation systems. On the other hand, larger amounts of data on the web not only result in more training data, but also in more data that potentially needs to be translated, either for gisting purposes (e.g. blogs, forum entries, online shop ratings) or for the purpose of publication (e.g. newspaper articles). In that respect, diverse text sources pose a particular challenge to machine translation systems which are usually tuned on small development sets to match the style of a particular target domain.

It is well-known in the machine translation literature that the type and amount of training data for a machine translation system has a significant impact on translation quality (Haddow and Koehn, 2012). Similarly, the performance of the same system applied to different types of test data can vary significantly. The issue of a translation system trained or tuned on one domain and applied to another domain is generally know as *domain mismatch*. The term subsumes several issues related to translation quality, such as *out-of-vocabulary* words, *grammatical constructions* and *lexical se-*

lection. Issues regarding lexical selection can be further divided into *style* issues and *word sense* issues. Style issues can occur when the translation system finds a correct translation that is expressed in either a more formal or a more colloquial style than appropriate for the target domain. For example, a translation system that was trained on European Parliament Proceedings would struggle to faithfully translate movie subtitles. Word sense issues often occur when the most frequent sense for a source word or phrase differs between training and target domain, resulting in the system picking a translation that does not reflect the intended sense in the target domain. The majority of words have more than one possible translation in any target language which makes the translation process extremely ambiguous.

Word sense problems in translation are due to the fact that most written languages have *homographs* - words with the same spelling but different meaning (*homonyms*) or words with the same spelling and different but semantically related meaning (*polysemes*) (Jurafsky and Martin, 2008). When translating between closely related languages, there are cases where words have different senses but more than one sense translates to the same target word. For example, the French word *état* and the English word *state* are both homonyms but they share two of their multiple senses: the political sense as in *L'État, c'est moi!* and *the Federal State of Germany*, and the situational sense as in *l'état des finances* and *the state of affairs*. In cases like these, the ambiguity of the words *état* and *state* does not have to be resolved during translation, but can be preserved in both translation directions. However, in many other cases where this is not possible the senses of the source words have to be disambiguated before translation.

The problem of *domain mismatch* is related to the issue of *homonyms* and *polysemes* in that multiple possible translations for the same source are not distributed uniformly across parallel training texts. Some translations are a lot more frequent in some text types than in others and depending on what data a translation system is trained on, the distributions over translations in the final MT system can vary significantly. These non-uniform distributions over translations are not (necessarily) the result of unrepresentative text samples, they are rather a symptom of the inherent context-dependence of translation. A training corpus can represent a particular domain - or more generally speaking - a *thematic* or *stylistic context* in which certain translations are more likely than others. Tying this dependency to a context rather to a domain lets us look at the problem in more general terms. A domain can be the context in which a specific translation is likely, but the context can also be defined as a document, a paragraph or a sentence. Deciding which contextual *scope* is necessary or sufficient for disambiguat-

ing word senses for the purpose of machine translation is a question that has received only limited attention in the literature (one example is Foster et al. (2010b)). While we remain far from fully answering this question, we hypothesize that it depends on each individual word type.

With the aforementioned issues in mind, a big challenge in the field of statistical machine translation is to build translation systems that are maximally suitable for a given *type* of input text or for a given, specific input text. The subfields of machine translation that address these issues are *domain adaptation* and *topic adaptation* and they differ in the amount of information that they assume as given for training and test data.

1.1 Motivation and goals

The main motivation behind this thesis is to address issues of lexical choice regarding *word senses* that arise when translation systems suffer *domain mismatch*. In trying to identify the appropriate *domain* or *topic* of a given text under translation, we seek to implicitly perform *word sense disambiguation* in order to select the appropriate translation given the underlying *word sense*. Figure 1.1 shows a motivating example for the problem of lexical selection based on the underlying word senses. The ambiguous French word *noyau* requires a different translation in English, depending on whether it is used in its scientific ($\rightarrow$ *nucleus*), economic ($\rightarrow$ *core*) or technical/IT ($\rightarrow$ *kernel*) sense. The automatic translations provided by GoogleTranslate[1] of the word *noyau* in its sentence context are wrong in two out of three cases, confirming that lexical selection is a non-trivial task for ambiguous - in this case homonymous - words. A human translator may have inferred the underlying word sense from the contextual information provided by the surrounding *sentence context* and then have selected the correct translation based on this word sense. However, each of these sentences was extracted from its surrounding *document context* and it is possible that even a human translator would have required wider contextual information in order to confidently identify the intended word sense.

Assuming a scenario where the *thematic* and *stylistic context* is not known advance to the translation system, we seek to answer the following research questions in this thesis:

[1] `https://translate.google.com`

Input	Il suffit d'éjecter le *noyau* et d'en insérer un autre, comme ce qu'on fait pour le clônage.
GoogleTranslate	Just eject the core and insert another, as is done for the cloning.
Reference	You can just pop out the **nucleus** and pop in another one, and that's what you've all heard about with cloning.
Input	Pourtant ceci obligerait les contribuables des pays de ce *noyau* à fournir du capital au sud.
GoogleTranslate	Yet this would require taxpayers in this **core** to provide capital to the south.
Reference	But this would unfairly force taxpayers in the **core** countries to provide capital to the south.
Input	Le *noyau* contient de nombreux pilotes, afin de fonctionner chez la plupart des utilisateurs.
GoogleTranslate	The nucleus contains many drivers to run in most users.
Reference	The precompiled **kernel** includes a lot of drivers, in order to work for most users.

Figure 1.1: Example of the ambiguous French source word *noyau* in three contexts, requiring three different translations due to the underlying word senses, as shown in bold in the reference translations. The automatic translations were produced with the online translation engine at `https://translate.google.com/#fr/en/` on July 7th, 2014.

- How can we represent contextual information in a way that helps a translation system improve its lexical selection?

- What is the amount of adaptation required to overcome the bias caused by the properties of the baseline model and the selected training data?

- What is the amount of contextual information required to confidently disambiguate word senses that underly the task of lexical selection?

- What is the relationship between *domain adaptation* and *topic adaptation* and how can their respective strengths be combined to yield a system of better quality and efficiency?

1.2 Modelling assumptions and evaluation

Throughout this thesis, we work within the framework of phrase-based translation models. This choice was primarily dependent on the language pairs used, for which the phrase-based models provide good baselines. Since we are targeting improvements in lexical selection, the particular translation framework would not be expected to have a significant influence on the ability to improve over the baseline systems. While for the first set of experiments we worked on models for German-English and English-French translation (both were language pairs in the IWSLT 2012 evaluation in which we participated), we subsequently switched to French-English for the remaining chapters. This language pair was chosen because phrase-based translation systems can usually achieve good baseline performance which makes it more feasible to attempt to make more fine-grained improvements concerning lexical choice. For example, there are no particular issues with long-range reorderings like there are for German-English, which can lead to problems during translation. The translation direction from French to English was chosen because the author is more fluent in English than in French while being able to read and understand both languages. This facilitates manual evaluation which is an important factor in the qualitative assessment of changes to the translation output.

While language model adaptation can be very powerful to improve system performance, in this work we focus mostly on translation model adaptation. We are interested in the changes in bilingual translational equivalence that occur when the context changes rather than just the probabilities of target words in a given context. The translation model effectively proposes high-probability hypotheses while the language model selects from the given set of hypotheses. Both models work closely together in search and each helps to shape the search space. It would be interesting to combine topic adaptation of both the translation and language model within the same system, however, for lack of readily available software for topic adaptation of language models, we leave this for future work. We do, however, explore combining topic-adapted translation models with domain-adapted language models.

For the sake of rapid experimentation and analysis, most of our adaptation work is carried out on context-specific phrase tables which we reload at context boundaries using a wrapper around the MT decoder.

Automatic evaluation of translation output is often an issue when targeting specific phenomena in translation. While large improvements in lexical choice can be detected

by automatic metrics that compare unigram or higher-order ngram matches, it can be useful to carry out more focused evaluation that targets the specific phenomenon we seek to improve. We use BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005) as the main metrics to evaluate our models as well as the WADE framework (Irvine et al., 2013) for more detailed analysis. WADE is an evaluation framework based on word alignments between source and reference sentences.

We also provide specific examples of translation output in each chapter. These examples were selected to illustrate the behaviour of a given system in translating ambiguous words when a systematic difference in translation output was found between two or more systems.

1.3 Structure of this thesis

In this thesis, we address both domain adaptation and topic adaptation with the primary focus on topic adaptation in order to improve *lexical selection*. In particular, we aim to automatically identify the topical structure in parallel training data that will help to find better translations in a given context. Even though the standard statistical translation systems make use of contextual information to some extent, we will show that there is still a lot to be gained from modelling contextual information in a more principled way. Another important aspect of the work presented here is that the goal of our approach to adapting a translation system is to deal with test inputs of unknown origin. This means that we want our system to be able to receive a random text from the web as input and adapt its translation model to that text by taking into account contextual information. This is a challenging task because it means that we cannot tune our system towards a particular target domain, as is the more common scenario of domain adaptation to date.

In the following chapters we discuss several approaches to machine translation that take into account structured context information. We start by reviewing relevant background literature related to statistical machine translation, topic modelling and contextual adaptation in Chapter 2.

In Chapter 3, we first explore discriminative training with sparse domain and topic features as a way to bias a translation model towards a domain or specific topic. We will show that this paradigm is not optimal for the problem we want to solve, mostly due to sparsity issues and the lack of extensibility. While yielding promising results for domain adaptation, the results for topic adaptation are less convincing. Therefore,

we subsequently turn to generative models for topic adaptation.

Following the "one sense per discourse" hypothesis (Gale et al., 1992), in Chapter 4 we design a model that induces latent topics over phrase pairs at the document level. The assumption is that the distribution over translations for a given source unit can change at document boundaries but remains constant within the same document. This enables us to adapt translation probabilities and other dynamic features to specific test documents.

In Chapter 5, we relax our assumption about document boundaries and turn to adaptation at the more fine-grained sentence level. Here, we follow the "distributional hypothesis" (Harris, 1954) that words occurring in similar contexts have similar meaning and design a model that measures the similarity of translation units and test contexts according to their distributional profiles.

We extend this model in Chapter 6 by inferring both local and global contextual information and comparing different combination methods.

In Chapter 7, we explore the relationship between domain adaptation and topic adaptation. We argue for an efficient architecture that combines both approaches, using learned topic representations of domains to automatically predict the domain of a given test document.

In Chapter 8, we summarise our findings and discuss possible directions for future work.

Orthographic conventions

We mark *emphasis* in text in italics and distinguish *meta usage* of words with slanted Roman style. When comparing the output of different translation systems to the input and reference sentences, we typeset the *input words* of interest in slanted Roman style, an incorrect translation with an underline and a **correct** translation as well as the **reference** translation in bold print.

1.4 Main contributions

The work presented in this thesis contributes to the field of statistical machine translation in the following ways. First, we test the framework of discriminative training with sparse features for domain and topic adaptation and conclude that the feature space becomes too sparse to be effective for topic adaptation. We then introduce a new bilin-

gual topic model that takes the entire document context into account and for the first time directly estimates topic-dependent phrase translation probabilities in a Bayesian fashion. We demonstrate its ability to improve over several domain adaptation baselines and further provide evidence for the advantages of bilingual topic modelling for SMT over the more common monolingual topic modelling. We introduce another new topic model for SMT which exploits the distributional nature of phrase pair meaning. Using this model, we explore combinations of local and global contextual information and demonstrate the utility of different levels of contextual information, which had not been previously examined for SMT. Finally, we shed light on the relationship between domain adaptation and topic adaptation and bring the two fields together by proposing to combine them in a framework that entails automatic prediction of domain labels at the document level.

The software needed to train and integrate all of the above adaptation models was developed as part of this thesis. All baseline systems were trained using the software in the Moses statistical machine translation system[2].

1.5 Publications

The work described in Chapter 3 has been previously published in the Proceedings of the International Workshop on Spoken Language Translation (IWSLT) (Hasler et al., 2012b). Part of that work was also used in a shared task submission at the same venue (Hasler et al., 2012a). The implementation of the online learning algorithm used for this work was published in the Prague Bulletin of Mathemetical Linguistics (PBML) as part of the Moses toolkit (Hasler et al., 2011).

The work decribed in Chapter 4 has been published in the Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL) (Hasler et al., 2014a) and part of the chapter overlaps with that publication.

Part of Chapter 5 and part of Chapter 6 were published in the Proceedings of the 9th Workshop on Statistical Machine Translation (WMT) (Hasler et al., 2014c). A variant of the model described in Chapter 5 was used for a shared task on translating L1 fragments in L2 context and is published in the Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval) (Hasler, 2014).

An extended version of Chapter 7 was published in the Proceedings of AMTA (Hasler et al., 2014b).

[2] http://www.statmt.org/moses/

Chapter 2

Statistical Machine Translation, Text Structure and Context

2.1 Statistical machine translation

In this section, we describe the framework of the statistical machine translation systems that are used for this thesis. We give a brief overview of different types of translation models and motivate our use of the phrase-based MT system. We also discuss evaluation metrics and their suitability to measure the changes in translation quality that we are interested in.

2.1.1 Types of translation models

Word-based models Early work on statistical machine translation by a research group at IBM introduced word-based translation models (Brown et al., 1990, 1993). In these models, word alignments and word-to-word translation probabilities are learned simultaneously using bootstrapping from sentence-aligned parallel corpora. Brown et al. (1993) define a series of models of increasing complexity, the IBM Models 1 to 5, and learn them by starting with the simplest model, using the learned parameters to initialise the model of the next higher complexity and so on up to the most complex model.

Because translating words in isolation has obvious drawbacks, the *noisy channel model* (Shannon, 1948) which stems from the field of information theory was applied to machine translation to combine the translation model with a language model to improve the fluency of translation output and introduce a form of context dependence

on the target side. Using Bayes rule, a simple machine translation model can be defined as

$$\begin{aligned} \operatorname*{argmax}_{\mathbf{e}} P(\mathbf{e}|\mathbf{f}) &= \operatorname*{argmax}_{\mathbf{e}} \frac{P(\mathbf{f}|\mathbf{e})P(\mathbf{e})}{P(\mathbf{f})} \\ &= \operatorname*{argmax}_{\mathbf{e}} P(\mathbf{f}|\mathbf{e})P(\mathbf{e}) \end{aligned} \tag{2.1}$$

where $\mathbf{f}$ is a source language sentence and $\mathbf{e}$ is a target language sentence. The denominator can be dropped from the equation because the probability of the source sentence is constant for all translations of the same source sentence.

2.1.2 Log-linear models

An extension to the noisy channel model that simplifies the integration of additional model components into the translation system is the *log-linear model* (Och and Ney, 2002). It is defined over feature functions $h(x)$ with weights λ and has the following form:

$$P(x) = \frac{\exp(\sum_{i=1}^{n} \lambda_i h_i(x))}{Z} \tag{2.2}$$

where Z is a normalisation constant that turns the numerator into a probability distribution. In the simple case that our model contains only the two features of the noisy channel model, two feature functions are defined as

$$h_1(x) = \log P(\mathbf{f}|\mathbf{e}) \tag{2.3}$$

$$h_2(x) = \log P(\mathbf{e}) \tag{2.4}$$

with associated weights λ_1 and λ_2. The structure of this model poses no limitations on the number and type of the feature functions which makes the model easily extensible. Another advantage is the ability to include feature weights and automatically train them to optimise translation quality over a set of sentence pairs:

$$\lambda_1^n = \operatorname*{argmax}_{\lambda_1^n} \{ \sum_{s=1}^{S} \log P_{\lambda_1^n}(\mathbf{e}_s|\mathbf{f}_s) \} \tag{2.5}$$

Phrase-based models Several extensions have been proposed to the word-based translation models that aim to fix some of the problems of translating words in isolation (Koehn et al., 2003; Och and Ney, 2004). Extending the translation units from word pairs to phrase pairs enables capturing local dependencies between words and offers an easy solution to incorporating one-to-many mappings between source and target language. Phrase pairs can also capture some of the syntactic phenomena of language that tend to group words together, for example into noun phrases. Other phenomena such as reordering during translation can be dealt with more easily at the phrase level than at the word level. The models of Koehn et al. (2003) apply heuristics to extract multi-word pairs from symmetrised word alignments in both translation directions, while the models of Och and Ney (2004) are based on alignment templates that generalise words to automatically learned word classes. Both types of models have been shown to outperform word-based models and still constitute the state-of-the-art for many language pairs. However, while using more contextual information than word-based models, phrase-based models still use only very local contextual information and ignore most of the contextual information in the input when selecting target phrases.

Syntax-based models Work by Yamada and Knight (2001), Chiang (2007) and others introduced translation models that deal with syntactic issues more explicitly than phrase-based models. For example, Yamada and Knight (2001) use parse trees on the source side to model reordering and insert case markers depending on the target language in order to improve the syntactic wellformedness of the output. Chiang (2007) propose a model based on *hierarchical* phrases that can be discontinuous and thereby enable long-distance reorderings between languages with different syntactic structure. While being linguistically more motivated in terms of some of the phenoma that are important for translation, these models do not fix the problems of the phrase-based system that are related to lexical selection.

Choice of translation framework In this thesis, we are interested in lexical selection for ambiguous words and phrases that depends on contextual information. Syntactic phenomena such as reordering are not the focus of this work and therefore we decided to carry out our experiments within the framework of phrase-based translation. The language pair used for most of our experiments, French-English, does not pose severe problems of long-distance reorderings and thus the phrase-based model is a reasonable baseline system for this language pair. However, a weakness of the phrase-based model

is its limitation to local source side context within the same phrase. This is often not sufficient for word disambiguation and informed lexical selection which is the main focus of this thesis.

2.1.3 Evaluation metrics

Evaluating the quality of machine translation output is a very difficult task because for each source sentence, there is an exponential number of correct translations in each target language, even though many of them may not be favoured by human evaluators. Human evaluation of translation output is probably the most reliable method of evaluation but very expensive to obtain and not practical to use throughout the development cycle of a translation system. Therefore, a more practical alternative are automatic *evaluation metrics* that require very little time and cost.

Two important criteria in measuring translation quality are *adequacy* and *fluency* and they are covered to varying extent by different metrics. Adequacy is a measure of how well a translation preserves the meaning of the source sentence while fluency measures the quality of the output as a fluent, natural-sounding sentence in the target language.

BLEU score The most popular metric is the BLEU score (Papineni et al., 2002) which computes ngram overlap of the output with a reference translation. Usually, the score is computed by taking the geometric average of different orders of ngram precision, i.e. the ratio of correct ngrams to the number of produced ngrams for each n. Because taking only precision into account would reward short translation with missing words but high ngram precision, the BLEU score includes a *brevity penalty* (BP). The penalty is computed as the exponentiated deviation from the length of the reference translation and is applied whenever the output contains fewer tokens than the reference translation. The final score is computed as

$$\text{BLEU-4} = BP \times \exp\Big(\sum_{i=1}^{4} w_i \log \text{ precision}_i\Big) \tag{2.6}$$

$$BP = \begin{cases} 1 & \text{if } |\text{output}| \geq |\text{reference}| \\ e^{1-\frac{|\text{reference}|}{|\text{output}|}} & \text{else} \end{cases} \tag{2.7}$$

where w_i are interpolation weights that are usually set to $\frac{1}{4}$. An alternative to the brevity penalty would be to incorporate *recall* into the metric. However, BLEU was

designed to enable scoring against multiple reference translations and it is unclear how recall should be defined in the presence of more than one reference translation.

METEOR score The lack of a notion of recall is one of the defects in the BLEU score that the METEOR[1] score (Banerjee and Lavie, 2005) tries to improve upon. The METEOR score requires finding a one-to-one/zero alignment between output and reference words and computes unigram precision and unigram recall over this alignment. It supports the use of different modules for string matching: *exact* computes matches using surface forms, *stem* applies the Porter stemmer to all words before computing matches, *synonym* maps two strings if they are synonyms according to WordNet, *paraphrase* supports the use of paraphrase tables for matching synonyms.

For this thesis, METEOR scores were computed using the *exact* and *stem* modules. We decided not to include synonym or paraphrase matching because of the possibility of errors or side effects in evaluating the translations of ambiguous words.

WADE framework WADE[2] (Irvine et al., 2013) is a framework for analysing machine translation output at the word level by computing word alignments between source and reference sentences. It distinguishes four different error types: *seen errors* occur when a source word was not seen in the training data, *sense errors* occur when the correct translation of a word is unseen, *score errors* occur when the correct translation was available but a different translation was chosen and *search errors* can occur because of pruning during beam search.

The class of errors we would be most interested in for the purpose of this thesis are *score errors* because they reflect the situation where a system can choose between several translations but may have an inappropriate distribution over the translations. Unfortunately, because the adapted phrase tables of our proposed models do not always match the number of entries in the baseline phrase table for a given sentence or document, the fine-grained WADE evaluation is not reliable for our models. However, WADE also provides functionality for restricting the word alignments to user-specified subsets. This allows for interesting insights because we can evaluate overall translation error rates grouped by part-of-speech or other criteria that seem suitable to distinguish classes of input words.

[1] Metric for Evaluation of Translation with Explicit Word Ordering
[2] Word Alignment Driven Evaluation

2.2 Topic modelling

In this section, we introduce Latent Dirichlet Allocation, which is the basis for the models developed in Chapter 4 and Chapter 5, as well as earlier versions of semantic models of text, Latent Semantic Analysis and Probabilistic Latent Semantic Analysis. All of these models are based on unsupervised machine learning techniques and learn latent structure in document collections from data. These models have been used for a range of natural language applications in the past, the first being information retrieval (Deerwester et al., 1990).

2.2.1 Latent Semantic Analysis

Latent Semantic Analysis (LSA) was introduced by Deerwester et al. (1990) as a means of reducing the dimensionality of vector space models, in particular vector space representations of text documents. Although it was first used as a retrieval model taking advantage of implicit higher-order structure between terms and documents, its ability to abstract from the original term-by-document matrix was found useful for other tasks as well.

LSA uses Singular Value Decomposition (SVD) by which a term-document matrix is decomposed into a set of orthogonal factors and the product of three other matrices. The original matrix can be reconstructed approximately by linear combination of these three matrices. The singular value decomposition of a term-document matrix X^{txd} can be written as

$$X^{t\times d} = T_0 S_0 {D_0}^{\top} \tag{2.8}$$

such that $T_0^{t\times m}$ and $D_0^{d\times m}$ have orthonormal columns and $S_0^{m\times m}$ is diagonal. Here, m denotes the rank of X. T_0 and D_0 and the matrices of left and right *singular vectors* and S_0 is the diagonal matrix of *singular values*. If the singular values are ordered by size, the first k values can be kept and the remaining set to zero. The reconstructed matrix $\hat{X}$ will then be of rank k. If S_0 is reduced to the matrix that contains only the non-zero values and the corresponding columns are deleted from T_0 and D_0, yielding $S^{k\times k}$, $T^{t\times k}$ and $D^{d\times k}$, the reduced model is

$$X^{t\times d} \approx \hat{X}^{t\times d} = TSD^{\top} \tag{2.9}$$

Deerwester et al. note that the choice of k is critical but an open issue in the lit-

erature. It should be chosen large enough to "fit all the real structure in the data, but small enough so that we do not also fit the sampling error or unimportant details." The lower-dimensional representations can be used, for example, to compare two documents. The hope is that even if the documents do not have any terms in common, they will be similar in the semantic space. The dot product between two column vectors in the matrix $\hat{X}$ expresses the similarity between documents. This document-to-document matrix can be written as

$$\hat{X}^\top \hat{X} = DS^2D \tag{2.10}$$

using the lower-dimensional document-factor matrices D. For the purpose of retrieval, dot products between query and document vectors can be used to measure similarity.

2.2.2 Probabilistic Latent Semantic Analysis

Hofmann (1999) introduce probabilistic Latent Semantic Analysis (pLSA) as a theoretically sound improvement over LSA which defines a generative model of the data. The starting point is the *aspect model*, a latent variable model for co-occurrence data that relates each type of observation (words and documents) to unobserved class variables $z_i, \ldots, z_K$. Under this model, the joint probability of a document d and a word w is given by one of two possible parameterisations

$$P(d,w) = P(d) \underbrace{\sum_{z \in Z} P(w|z)P(z|d)}_{P(w|d)} \tag{2.11}$$

$$P(d,w) = \sum_{z \in Z} P(z) \underbrace{P(d|z)P(w|z)}_{P(d,w|z)} \tag{2.12}$$

This introduces a conditional independence assumption between documents and words: they are generated independently of each other given the latent variable z. The conditional distributions $P(w|d)$ for all documents are approximated by multinomials $P(w|z)$. The parameters of the aspect model can be trained with the EM algorithm, using the parameterisation in Equation 2.12.

Hofmann shows how LSA and pLSA relate to each other. The aspect model can be parameterised in matrix notation

$$D = (P(d_i|z_k))_{i,k} \tag{2.13}$$

$$T = (P(w_j|z_k))_{j,k} \tag{2.14}$$

$$S = diag(P(z_k))_k \tag{2.15}$$

$$X = DST^\top. \tag{2.16}$$

The mixture components in pLSA correspond to the K factors in LSA and the mixture proportions in pLSA substitute the singular values in LSA. The crucial difference is the objective function with which the decomposition in LSA and the approximation in pLSA are determined. In addition, the mixture approximation in Equation 2.12 and Equation 2.16 is a probability distribution which makes it much more interpretable than the singular values of LSA.

A problem with pLSA is that it learns topic mixtures $P(z|d)$ only for the documents in the training data, as pointed out by Blei et al. (2003). This means that the model does not easily generalise to unseen documents and that the number of parameters grows with the size of the training set. The number of parameters the model has to learn is $KV + KM$ (K distributions over the vocabulary V and M topic mixtures where M is the size of the document collection).

2.2.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a probabilistic, generative model of document collections introduced by Blei et al. (2003). It represents documents as random mixtures over *latent topics* and each topic as a multinomial over the words in the vocabulary[3]. These multinomial distributions can be seen as unigram language models, where each model assigns higher probability to particular words while all other words still have some probability of being generated. The generative story of a document collection according to LDA is the following:

For each document d in the collection:

1. Choose the length of the document $N \sim$ Poisson(ζ)
2. Choose a topic mixture $\theta \sim$ Dirichlet(α)

[3]LDA is also referred to as an *admixture model*, a term that refers to a mixture whose components are itself mixtures.

3. For each of the N words w_n:

   (a) Choose a topic $z_n \sim$ Multinomial(θ)

   (b) Choose a word w_n from $P(w_n|z_n)$, also denoted as the topic-specific multinomial ϕ_z, $w_n \sim$ Multinomial(ϕ_z)

α and β are the concentration parameters of the Dirichlet priors over topic and word distributions, respectively. The probability of generating a document, that is, the bag-of-words given by the document, is given by

$$P(\mathbf{w}|\alpha,\beta) = \int P(\theta|\alpha)\Big(\prod_{n=1}^{N}\sum_{z_n} P(z_n|\theta)P(w_n|z_n,\beta)\Big)d\theta. \tag{2.17}$$

For each position in the document, the probability of the latent variable z is summed out and the probabilities of generating each word are assumed to be independent given topic z and therefore multiplied. Equation 2.17 entails the *exchangeability assumption* about words in a document which means that the order in which they appear can be neglected. A similar assumption is made for documents in the collection.

The probability of the topic distribution θ is the probability of a point in the (K-1) simplex[4] and is integrated out. This probability is given by the Dirichlet distribution, which is a probability distribution over multinomial distributions:

$$Dirichlet(\alpha_1,\ldots,\alpha_K) = P(\theta|\alpha) = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_k^{\alpha_k-1} \tag{2.18}$$

The hyperparameters α_k can be interpreted as pseudo-counts of the quantities that the Dirichlet distributions describes. For a Dirichlet distribution over topic mixtures, α_k is a pseudo-count of topic k in a document which is a measure of how often we expect to see the topic in a document, prior to having seen any words from the document. Similar to the prior distribution over document topics, a prior is placed on the word multinomials in order to smooth the word counts[5].

The model formulation of LDA improves over pLSA in that it defines a probabilistic model at the level of documents. Instead of estimating a large set of individual parameters for all training documents, LDA treats topic mixtures as random variables with a Dirichlet prior. The number of parameters the model has to learn is $KV + K$

[4]The (K-1) simplex represents all possible multinomial distributions over K discrete random variables.

[5]This is particularly important when applying the learned model to new documents which may contain unseen words.

(K distributions over the vocabulary V and a K-parameter hidden random variable for topic mixture weights) and does not grow with the size of the document collection.

The learning problem for LDA amounts to learning the conditional distribution of the topic structure in a document collection given the observed documents. This is expressed by the posterior distribution over the hidden variables given the observed variables,

$$P(\mathbf{z},\theta,\phi|\mathbf{w},\alpha,\beta) = \frac{P(\mathbf{z},\mathbf{w},\theta,\phi|\alpha,\beta)}{\sum_{\mathbf{z}} P(\mathbf{z},\mathbf{w}|\alpha,\beta)}. \tag{2.19}$$

The joint probability distribution in the numerator of Equation 2.19 can be written as a product of prior and likehood terms

$$\begin{aligned} P(\mathbf{z},\mathbf{w},\theta,\phi|\alpha,\beta) &= \underbrace{P(\theta|\alpha)P(\mathbf{z}|\theta)}_{P(\mathbf{z},\theta|\alpha)}\underbrace{P(\phi|\beta)P(\mathbf{w}|\phi,\mathbf{z})}_{P(\mathbf{w},\phi|\mathbf{z},\beta)} \\ &= \underbrace{\prod_{d=1}^{D} P(\theta_d|\alpha)}_{\text{topic prior}} \underbrace{\prod_{d=1}^{D}\prod_{n=1}^{N} P(z_{d,n}|\theta_d)}_{\text{likelihood of topics}} \times \underbrace{\prod_{k=1}^{K} P(\phi_k|\beta)}_{\text{word prior}} \underbrace{\prod_{d=1}^{D}\prod_{n=1}^{N} P(w_{d,n}|\phi_{z_{d,n}})}_{\text{likelihood of words}} \end{aligned} \tag{2.20}$$

Writing out the specific distributions for the first part of Equation 2.20 (the second part can be derived similarly) yields

$$\begin{aligned} P(\mathbf{z},\theta|\alpha) &= \prod_{d=1}^{D} \frac{\Gamma(\sum_{k=1}^{K}\alpha)}{\prod_{k=1}^{K}\Gamma(\alpha)} \prod_{k=1}^{K} \theta_k^{\alpha-1} \prod_{d=1}^{D}\prod_{k=1}^{K} \theta_k^{c_{d,k}} \\ &= \prod_{d=1}^{D} \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K} \prod_{k=1}^{K} \theta_k^{\alpha-1+c_{d,k}} \end{aligned} \tag{2.21}$$

because of the conjugacy of the Dirichlet prior and the multinomial distribution. Integrating over the topic and word multinomial parameters in the joint distribution yields

$$\begin{aligned} P(\mathbf{z},\mathbf{w}|\alpha,\beta) &= \int\int P(\mathbf{z},\mathbf{w},\theta,\phi|\alpha,\beta)\,d\theta\,d\phi \\ &= \int P(\mathbf{z},\theta|\alpha)\,d\theta \times \int P(\mathbf{w},\phi|\mathbf{z},\beta)\,d\phi. \end{aligned} \tag{2.22}$$

By reformulating and dropping some of the terms (and plugging in the specific distributions as in Equation 2.21), the results of the two integrals are

$$P(\mathbf{z}|\alpha) \propto \prod_{d=1}^{D} \frac{\prod_{k=1}^{K} \Gamma(c_{d,k,.} + \alpha)}{\Gamma(\sum_{k=1}^{K} c_{d,k,.} + \alpha)} \tag{2.23}$$

$$P(\mathbf{w}|\mathbf{z}, \beta) \propto \prod_{k=1}^{K} \frac{\prod_{w=1}^{W} \Gamma(c_{.,k,w} + \beta)}{\Gamma(\sum_{w=1}^{W} c_{.,k,w} + \beta)} \tag{2.24}$$

Therefore, the probability of the observed and hidden variables is

$$P(\mathbf{z}, \mathbf{w}|\alpha, \beta) \propto \prod_{k=1}^{K} \frac{\prod_{w=1}^{W} \Gamma(c_{.,k,w} + \beta)}{\Gamma(\sum_{w=1}^{W} c_{.,k,w} + \beta)} \times \prod_{d=1}^{D} \frac{\prod_{k=1}^{K} \Gamma(c_{d,k,.} + \alpha)}{\Gamma(\sum_{k=1}^{K} c_{d,k,.} + \alpha)} \tag{2.25}$$

and Equation 2.19 (omitting hyperparameters) simplifies to

$$P(\mathbf{z}|\mathbf{w}) = \frac{P(\mathbf{z}, \mathbf{w})}{\sum_{\mathbf{z}} P(\mathbf{z}, \mathbf{w})} \tag{2.26}$$

This is intractable to compute because it involves enumerating an exponential number[6] of possible sequences of word-topic assignments. However, several algorithms have been proposed to approximate the posterior distribution, such as *variational inference* (Blei et al., 2003) and *Gibbs sampling* (Griffiths and Steyvers, 2004). Variational methods are a deterministic alternative to sampling-based methods. Instead of collecting samples to approximate the posterior distribution, variational methods transform the inference problem into an optimisation problem by defining a family of simpler distributions (usually by introducing independence assumptions) and finding the parameterisation that is closest to the desired posterior distribution.

We first give a brief introduction to Markov Chain Monte Carlo methods of which Gibbs sampling is an instance. We then discuss *collapsed Gibbs sampling* for LDA as well as *collapsed variational Bayes* which is a hybrid of variational inference and Gibbs sampling.

2.2.4 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a framework for sampling from complex probability distributions that are intractable to compute exactly. An example for such a complex distribution is the distribution over the space of topic assignments $\mathbf{z}$ in LDA. Even though the state of the hidden variables can be of interest itself, in many cases the posterior distribution over the hidden variables is used to compute expectations of

[6] There are K^n possible topic assignments where n is the number of tokens in the document collection.

functions that depend on the distribution over hidden states. Because it is often impossible to visit all states in a high-dimensional space, the normalisation constant for computing the distribution over states $\mathbf{z}$ cannot always be computed exactly. Sampling methods offer a way to approximate this distribution by replacing the sum or integral over all possible states by the sum over a small set of samples from $P(\mathbf{z})$. In order to collect such a set of samples, we need to explore the space in a way that most of the time is spent in high-probability regions (Bishop, 2006; Mackay, 2003).

For MCMC, given a high-dimensional state space with states $\mathbf{z}$, the sequence of explored states $\mathbf{z}^{(1)}$, $\mathbf{z}^{(2)}$, ..., $\mathbf{z}^{(M)}$ forms a Markov chain where the probability of visiting the next state only depends on the current state $\mathbf{z}^{(\tau)}$. This simplifies the problem of drawing samples from $P(\mathbf{z})$ but also means that the samples are not independent. At each step in the chain, a sample $\mathbf{z}^*$ is drawn from a so-called *proposal distribution* $Q(\mathbf{z}|\mathbf{z}^{(\tau)})$ and either accepted or rejected.

In the *Metropolis* algorithm (Metropolis et al., 1953), the acceptance criterion compares the unnormalised probabilities of the current and proposed next state

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = min\left(1, \frac{\tilde{P}(\mathbf{z}^*)}{\tilde{P}(\mathbf{z}^{(\tau)})}\right) \tag{2.27}$$

and accepts the next state with probability 1 if the value of $\tilde{P}(\mathbf{z})$ increases, and with probability A otherwise. Here, $P(\mathbf{z}^*) = \tilde{P}(\mathbf{z}^*)/Z_p$ where Z_p is the normalisation constant and crucially, the proposal distribution does not require computation of Z_p. We only need to be able to evaluate the unnormalised distribution $\tilde{P}(\mathbf{z})$. If the sample is accepted, it is added to the chain ($\mathbf{z}^{(\tau+1)} = \mathbf{z}^*$). Otherwise, the current state is duplicated and added to the chain ($\mathbf{z}^{(\tau+1)} = \mathbf{z}^{\tau}$). The sequence of states collected in this way is not independent though, because successive samples are highly *correlated*. Instead, if we collect only every M^{th} sample, we can obtain a set of uncorrelated samples for sufficiently large M.

The *Metropolis-Hastings* algorithm is a generalisation of the Metropolis algorithm where the proposal distribution does not have to be symmetric (for symmetric distributions, $Q(\mathbf{z}^*|\mathbf{z}^{(\tau)}) = Q(\mathbf{z}^{(\tau)}|\mathbf{z}^*)$). In the general case, the acceptance criterion for the next state is

$$A(\mathbf{z}^*, \mathbf{z}^{(\tau)}) = min\left(1, \frac{\tilde{P}(\mathbf{z}^*)Q(\mathbf{z}^{(\tau)}|\mathbf{z}^*)}{\tilde{P}(\mathbf{z}^{(\tau)})Q(\mathbf{z}^*|\mathbf{z}^{(\tau)})}\right). \tag{2.28}$$

2.2.4.1 Properties of the Markov chain

The goal of running an MCMC sampler is to reach convergence to the desired target distribution over the state space. This goal is reached when the distribution becomes *stationary* or *invariant*. In a first-order Markov chain, the probability of being in state $\mathbf{z}^{(m)}$ for $m \in 1,\ldots,M-1$ is defined as

$$P(\mathbf{z}^{(m+1)}|\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(m)}) = P(\mathbf{z}^{(m+1)}|\mathbf{z}^{(m)}) \tag{2.29}$$

subject to the independence assumption for all previous states except the current (Bishop, 2006). Thus, the distribution over the state space is invariant if transitioning to a new state does not change this distribution. Another requirement for the Markov chain to converge to the target distribution is *ergodicity* which means that each state must be reachable from any given initial state. This ensures that we can reach an area in the state space where it is possible to draw samples from the correct distribution. This can be achieved by ensuring that the transition probabilities are non-zero for all states.

2.2.4.2 Gibbs Sampling

Gibbs Sampling (Geman and Geman, 1984) is a special case of the Metropolis-Hastings algorithm where the proposal distributions Q are defined in terms of conditional probabilities, $Q_i(\mathbf{z}^*|\mathbf{z}) = P(z_i^*|\mathbf{z}^{(-i)})$, where only one variable in the state space is changed while all other variables are kept fixed. $\mathbf{z}^{(-i)}$ denotes the set of all topic variables except for the one that is being resampled. Such distributions are usually easy to sample from and thus the method allows to move around the state space by making steps in one direction at a time. Gibbs sampling is also known as *alternating conditional sampling* (Steyvers and Griffiths, 2007).

It can be shown that because of the properties of this specific kind of proposal distribution, the acceptance criterion for the Metropolis-Hastings algorithm simplifies to be equal to 1 for all proposed states (Bishop (2006), Equation 11.49). Therefore, all steps are accepted in Gibbs sampling. A single iteration of a Gibbs sampler involves resampling all variables in the state space once, which can be done in sequential or random order. Variables are assigned their resampled value immediately, so that each sample depends on the most recent state of all other variables.

2.2.5 Inference for Latent Dirichlet Allocation

Depending on the application, the quantities of interest for LDA are the probability distribution over the hidden variables given the words in all documents, $P(\mathbf{z}|\mathbf{w})$, as well as the topic-word distributions ϕ and the document-topic distributions θ. The first problem that needs to be solved is to estimate the posterior distribution over the state space $\mathbf{z}$ given the observed word tokens $\mathbf{w}$ which corresponds to the assignment of topics to all words tokens in the document collection. Again, this space can be very large and is often approximated by Gibbs sampling. When the distributions ϕ and θ are marginalised out, only the distribution over latent topics needs to be estimated and the algorithm is referred to as *collapsed* Gibbs sampling. With a set of samples from the posterior distribution over $\mathbf{z}$, ϕ and θ can then be estimated from the collected co-occurrence counts and pseudo-counts (Griffiths and Steyvers, 2004).

2.2.5.1 Collapsed Gibbs sampling for LDA

The Gibbs sampling procedure to infer the posterior distribution over the latent variables is relatively simple for the LDA model. Given an initialisation of topic assignments to word tokens[7], each word token in the document collection is visited in turn and its topic variable is resampled depending on the topic assignments of all other word tokens. The predictions are based on two matrices of topic-word and document-topic co-occurrence counts, respectively, and the counts are updated with each resampled variable. At each step, the sampler removes the counts involving the current topic assignment $z_{d,i}$ at position i in document d from the count matrices, resamples the assignment of $z_{d,i}$ and then updates the count matrices with the new topic assignment.

In order to estimate $P(\mathbf{z}|\mathbf{w})$ (Equation 2.26), we first need to compute $P(\mathbf{z},\mathbf{w})$ from which the multinomial distributions ϕ and θ can be marginalised out, leading to Equation 2.25. The conditional distribution needed for Gibbs sampling, the probability of a single topic assignment given all other topic assigments and the observed word tokens, can be derived from Equation 2.25 by solving for a single topic assignment $z_{d,i}$ and removing constant terms with respect to $z_{d,i}$:

$$P(z_{d,i} = k|\mathbf{z}^{(-d,i)},\mathbf{w}) \propto \frac{c^{(-d,i)}_{.,k,w_{d,i}} + \beta}{c^{(-d,i)}_{.,k,.} + W\beta} \times \frac{c^{(-d,i)}_{d,k,.} + \alpha}{c^{(-i)}_{d,.,.} + K\alpha} \tag{2.30}$$

[7] Initialisation can be performed randomly or by using an online version of the sampler that takes the partial topic assignments into account.

This distribution is normalised by summing over the values computed for all possible topics and dividing by the sum. Intuitively, the first ratio expresses the probability of a word w under topic k, while the second ratio expresses the probability of topic k occuring in document d (Griffiths and Steyvers, 2004). The denominator of the second ratio is constant with respect to $z_{d,i}$ and can therefore be neglected.

2.2.5.2 Collapsed Variational Bayes for LDA

Because inference in models like LDA often needs to be carried out on large data sets, Teh et al. (2006) propose *collapsed variational Bayesian inference* (CVB) as an alternative inference method that is computationally more efficient than Gibbs sampling, achieves better performance than *variational Bayesian inference* (VB) and similar performance to *collapsed Gibbs sampling*.

In variational Bayesian inference, the parameters θ and ϕ and the latent variables $\mathbf{z}$ are assumed to be independent which can lead to results that are far from the true posterior distribution. However, it is simple to implement using a set of update equations that converge to a local minimum of the negative log likelihood.

Collapsed Gibbs sampling has been shown to converge quickly which is partially attributed to the fact that there is only a weak dependency between $z_{d,i}$ and $\mathbf{z}^{(-d,i)}$ (Teh et al., 2006). This can be seen from Equation 2.30 because $z_{d,i}$ only depends on other topic assignments through the count variables. However, it is difficult to diagnose when the sampler has converged and we may need to collect a large number of samples to reduce noise.

Combining the advantages of both, CVB is a variational algorithm which does not assume independence between parameters and latent variables but models the dependence in an exact fashion. It only assumes independence between latent variables, which as argued before are only weakly dependent on each other. The posterior over latent variables and parameters is approximated as

$$\hat{q}(\mathbf{z},\theta,\phi) = \hat{q}(\theta,\phi|\mathbf{z}) \prod_{d,i} \hat{q}(z_{d,i}|\hat{\gamma}_{d,i}) \tag{2.31}$$

with variational parameters $\hat{\gamma}_{d,i}$. Omitting details, the distribution for a single variable $z_{d,i}$ is computed as

$$\hat{\gamma}_{d,i,k} = \hat{q}(z_{d,i}=k) \propto \exp\left(\mathbb{E}_{\hat{q}(\mathbf{z}^{(d,i)})}\left[log\left(c^{(d,i)}_{d,k,.}+\alpha\right)+log\left(c^{(d,i)}_{.,k,w_{d,i}}+\beta\right)\right.\right.$$
$$\left.\left.-log\left(c^{(d,i)}_{.,k,.}+W\beta\right)\right]\right) \quad (2.32)$$
$$\propto \left(\mathbb{E}_{\hat{q}}[c^{(d,i)}_{d,k,.}]+\alpha\right)\left(\mathbb{E}_{\hat{q}}[c^{(d,i)}_{.,k,w_{d,i}}]+\beta\right)\left(\mathbb{E}_{\hat{q}}[c^{(d,i)}_{.,k,.}]+W\beta\right)^{-1} \quad (2.33)$$

where the second line is derived from the first line by approximating $\mathbb{E}_{\hat{q}}\left[\log\left(c^{(d,i)}_{d,k,.}+\alpha\right)\right] \approx \log(\mathbb{E}_{\hat{q}}[c^{(d,i)}_{d,k,.}]+\alpha)$ for each of the three expectations, respectively. This is very similar to Equation 2.30 for Gibbs sampling except that the counts (also referred to as *fields*) are replaced by their means. The expectation of the counts is computed with a first-order Gaussian approximation

$$\mathbb{E}_{\hat{q}}[c^{(d,i)}_{d,k,.}] = \sum_{i' \neq i} \hat{\gamma}_{d,i',k} \quad (2.34)$$

for $c^{(d,i)}_{d,k,.}$ and analoguously for the other counts. The reasoning is that because each count variable represents a sum over a large number of Bernoulli variables (with outcome 1 if $z=k$ and 0 otherwise), it can be approximated by a Gaussian. The mean of the Gaussian is given by the sum of the means of the individual Bernoulli variables, $\hat{\gamma}_{d,i',k}$.

In summary, inference using CVB requires repeatedly updating the distributions over topics for each word token using Equation 2.33, removing and adding expectations of counts in the same way as removing and adding counts in Gibbs sampling, until the distributions converge. Being a deterministic algorithm[8], it does not require a burn-in phase like Gibbs sampling and the final state of the converged distributions can be used for further computations.

2.3 Overview of related work in SMT

In this section, we review the relevant literature for domain and topic adaptation in machine translation as well as other related work on context-dependent translation and word sense disambiguation for MT. Where possible, we follow a chronological order in describing work on the various subtopics.

In most of the literature, the term *domain* is defined as the source of a corpus and we will adopt this definition. For example, the term *news domain* refers to a text collection

[8]The only non-deterministic part of the algorithm is the initial random assignment of topic distributions to all documents in the collection.

that has been extracted from news websites, while the term *medical domain* refers to a collection of documents from the European Medicines Agency. The term *topic* on the other hand is defined in terms of the latent variables in a topic model (Section 2.2.3).

2.3.1 Context dependence and word sense disambiguation

Most standard SMT systems employ only a limited amount of contextual information, in the form of source context within the same phrase pair and target language model context (typically a history of four words). This section provides a brief overview of work that tries to incorporate additional contextual information from the source sentence to improve rule selection and lexical selection.

In order to improve lexical selection, Carpuat and Wu (2005) use maximum entropy word sense disambiguation (WSD) classifiers with a feature set of source POS tags, context bag-of-words and collocations to predict Chinese WordNet-style senses. They integrate these word senses into MT but do not report improvements in translation quality. Using the first method, the decoder is forced to produce the translation that maps to the one-best gloss of the predicted sense. Using the second method, the predicted senses are used for automatically post-processing the output. Vickrey (2005) reformulate the classification task as predicting possible target translations rather than predefined senses. They show improved translation selection in a *blank-filling* task, which is a subtask of the full MT task focused on word translations. Subsequent work (Carpuat and Wu, 2007c; Chan et al., 2007) adopts this formulation and shows improved translation performance on the full MT task. Carpuat and Wu (2007a,b) expand word sense disambiguation to phrase sense disambiguation and show further improvements. Features in the target phrase classifiers are the context bag-of-words, local collocations, position-sensitive local POS tags and basic dependency features. Similarly, Gimenez and Marquez (2007) extend the work of Vickrey by moving to phrase translations and tackling full translation. Though not yielding any BLEU improvements, they show small improvements according to several other evaluation metrics including ROUGE and METEOR.

Ittycheriah and Roukos (2007) introduce Direct Translation Model 2 which employs syntactic information (POS tags) and context information (neighboring source words and previous two target words) within a maximum entropy model to predict the correct transfer rules. Similarly, He et al. (2008) improve syntactic rule selection with source side context features: one word (POS) to the left and right and boundary words

(POS) of a rule. Hasan et al. (2008) introduce triplet models for MT which use an additional target word trigger to estimate the probability of a source word, $p(f|e,e')$. The resulting probability distribution is used in reranking.

Gimpel and Smith (2008) use a set of local source phrase features to condition translations on the context: left and right context (word or POS tags) syntactic features (consistency, non-terminal labels, ...) and positional features (start/end of sentence, relative position and coverage). Feature selection methods showed that the local lexical and POS features worked best, depending on the language pair.

Shen et al. (2009) integrate the probability of the left and right source words given the target phrase as an additional language model score.

While all of the above methods use additional contextual information to improve translation, none of them makes use of context beyond the sentence level. Thus, each occurrence of the same word is modelled independently at the token-level. In contrast, Mei (2010) provide a study on the impact of missing contextual information in translation, considering three types of contextual information: *discourse coherence* information, knowledge of *topic* or *domain* and *real-world/multimodal* information. The authors note that

> [...], the relevant information for resolving a word sense distinction is often not located in the immediately surrounding context but is either at a more distant location in the discourse, or it is part of the participants' background knowledge. [...] Thus, [...] we propose to utilize unsupervised, *global* word sense disambiguation, in order to obtain better modeling of the topic and domain knowledge that is implicitly present in meeting conversations.

They employ an unsupervised WSD method that assigns a score to each possible target translation that depends on its similarity to other content word translation candidates in a given document. Small but consistent improvements in position-independent word error rate (PER) are demonstrated (though no improvements in BLEU). While the work shows promising results for *global* WSD, the approach is not compared to *local* WSD methods.

Apidianaki et al. (2012) integrate the prediction of a WSD classifier into a local unigram language model that is estimated for every test sentence. The WSD system consists of feature vectors of source word lemmas from the sentence contexts where a source word was translated to a particular target word. The classifier outputs a distribution over possible target words which is computed using association scores between the feature vectors of each translation and the feature vector of the test sentence context.

	Cross-domain	**Dynamic**
Corpus boundaries	domain adaptation*	dynamic domain adaptation
Automatic sentence clustering (train/dev)	unsupervised domain adaptation	unsupervised multi-domain adaptation
Document boundaries	topic adaptation	dynamic topic adaptation

Table 2.1: Overview of the different types of model adaptation for machine translation: rows denote the structuring of the training/development data, columns denote the adaptation scenario (*: common type of adaptation).

2.3.2 Relationship between domain and topic adaptation

Table 2.1 provides an overview of different domain and topic adaptation methods, grouped by the structuring of the training data and the adaptation scenario. We use the definition of *cross-domain* versus *dynamic* adaptation by Foster and Kuhn (2007):

> In *cross-domain* adaptation, a small sample of parallel in-domain text is available, and it is used to optimize for translating future texts drawn from the same domain. In *dynamic* adaptation, no domain information is available ahead of time, and adaptation is based on the current source text under translation.

A very common adaptation scenario is *cross-domain adaptation* with corpus boundaries and a common approach is *mixture modeling*, where models are built separately for each training corpus and mixture weights combining them are optimised for the target domain (top left of the table). When moving from cross-domain to *dynamic adaptation*, a global set of mixture weights is not sufficient. Instead, the mixture weights have to be adjusted for each test instance (which may be a document or a sentence), for example depending on automatic domain prediction.

When training corpora or test sets contain very diverse text, standard domain adaptation methods may not be a good fit. Instead, automatic sentence clustering can be used to find clusters of similar data either in the training data or in the development data. Since the latter induces clusters that potentially correspond to multiple target domains, it can be seen as unsupervised multi-domain adaptation. When the training data is diverse but document boundaries are given, topic models can be used to

find hidden structure in corpora which can be much more fine-grained than corpus or domain labels.

While for *cross-domain* adaptation, the relevance of each topic for the target domain can be determined on a development set, topic inference techniques can be used to infer the topical structure of unseen test documents for *dynamic* adaptation. Most work on topic adaptation for SMT focuses on the latter scenario.

2.3.3 Domain adaptation

Foster and Kuhn (2007) were the first to apply mixture modeling techniques to SMT domain adaptation and much of the following work in subsequent years builds on their ideas. Their work provides domain adaptation experiments that compare *cross-domain* versus *dynamic* adaptation, *linear* mixtures versus *loglinear* mixtures as well as translation model versus language model adaptation. They also compare different methods for assigning weights and the effect of adaptation at different structural levels (test set/genre/document). For all their experiments, they assume training data from different corpora for which separate component models are trained. For language modelling, a static global model is used in addition to the component models. In-domain tuning data (newswire) is assumed in case of cross-domain adaptation. For both cross-domain and dynamic adaptation, linear mixture weights are set according to distance metrics that measure the relation between a test text and each model component. The main findings can be summarised as follows:

- (uniform) *linear* mixture weights outperform *loglinear* weights set along with all other model components on a development set
- for *cross-domain* adaptation:
  - both translation model (TM) and language model (LM) adaptation improve performance
- for *dynamic* adaptation:
  - LM adaptation consistently improves over the baseline while TM adaptation does not

At the same time, Koehn and Schroeder (2007) report experiments on domain adaptation using separate in-domain and out-of-domain language models and multiple decoding paths for in-domain and out-of-domain phrase tables. The additional models

are combined log-linearly with the other models. While they report good results for the setup with multiple decoding paths, Foster and Kuhn (2007) observed better performance with a linear combination. Sennrich (2012b) use perplexity to learn interpolation weights for different model components[9] as proposed by Foster and Kuhn (2007) but optimise translation model perplexity on an in-domain development set. This method for setting linear interpolation weights for the translation model had been proposed earlier by Foster et al. (2010a) who used it for one of their baseline systems. Sennrich also optimise weights for each phrase table feature separately which seems to be slightly more robust than using a single optimised weight for all features.

Sennrich (2012a) propose an unsupervised variant of domain adaptation that does not rely on corpus boundaries but instead performs automatic clustering of the training data. Separate models are trained for each cluster and mixture weights are adjusted globally on a development set of the target domain.

A slightly different line of work discriminatively weights the training data depending on the performance on an in-domain development set. Matsoukas et al. (2009) propose corpus weight estimation to downweight certain parts of the training data. They learn to map sentences to weights using sentence-level feature vectors that encode collection and genre ids. Using the learned weights, the translation model can be estimated from weighted counts. Foster et al. (2010a) propose a more fine-grained approach that learns instance weights at the level of phrase pairs. Weights are learned for all phrase pairs in the out-of-domain training set according to features that measure their generality or similarity to the in-domain set. The estimated out-of-domain translation probabilities are then interpolated with in-domain probabilities. Bisazza et al. (2011) also make use of corpus identifiers and differentiate between data sources by using in-domain phrase table scores when available and backing off to out-of-domain scores when a phrase pair was not seen in the in-domain corpus. Chen et al. (2013b) propose an approach that is related to corpus weighting but defined in terms of a vector space model. Each phrase pair is represented by a vector where each dimension corresponds to one of the training corpora and a similar vector is defined for an in-domain development set, computed over all phrase pairs that can be extracted from that set. A similarity feature measures the vector similarity of each phrase pair in the phrase table to the development set and thereby favours phrase pairs that occur in similar corpora as the ones in the development set.

[9] A comparison of learning corpus weights instead of interpolation weights did not reveal a clear preference for either of the two.

Other approaches to domain adaptation have focused on finding additional, suitable training data (Daumé III and Jagarlamudi, 2011; Pecina et al., 2011), using monolingual in-domain data to produce synthetic training data (Schwenk, 2008; Bertoldi and Federico, 2009), ensemble decoding with in-domain and out-of-domain translation models (Razmara, 2012) and data selection techniques, for example Moore and Lewis (2010) for language model adaptation and Axelrod et al. (2011) for translation model adaptation. In terms of language model adaptation, interpolating in-domain and out-of-domain models using development set perplexity has become a fairly standard adaptation technique (Schwenk and Koehn, 2008). Chen et al. (2013a) first introduce domain adaptation of reordering models for MT.

In this thesis, we address translation model adaptation, but also compare to baseline systems with adapted language models.

2.3.3.1 Dynamic adaptation approaches

In comparison to cross-domain adaptation approaches for SMT, there is a smaller body of work addressing dynamic adaptation. A simple approach to dynamic translation model adaptation to a given test set was proposed by Hildebrand et al. (2005). For each sentence, a set of similar sentences (according to a vector space model with TF-IDF weights) from the training data is selected and added to the test-specific training set. The training set is used to build an adapted translation model for the test set. While the approach demonstrates that data selection can have a large impact on translation quality, the proposed method is impractical for adaptation at runtime.

Foster and Kuhn (2007) address cross-domain as well as dynamic adaptation, using distance metrics to compare a test text to each of the trained model components and set the mixture weights proportional to the similarity scores. However, while they show improvements over the baseline for dynamic language model adaptation, their approach did not yield improvements for dynamic translation model adaptation.

Finch (2008) employ probabilistic mixture weights between models that can change dynamically on a segment-by-segment basis. They differentiate between three model types, *general*, *questions* and *declarations*. The general translation model is used with a fixed weight during decoding while the sentence-type models are interpolated with mixture weights that depend on the output of a maximum entropy question classifier. While the approach is dynamic, it relies on a binary classifier that distinguishes between only two classes and assumes labelled training data.

Yamamoto and Sumita (2008) cluster the training data into sub-corpora and predict the cluster of each test sentence to combine a general and cluster-specific model according to a fixed weighting. Banerjee et al. (2010) assume that the training data has domain labels and use automatic domain classifiers to translate each test sentence with a domain-specific model.

Extending the work of Yamamoto and Sumita, Sennrich et al. (2013) perform multi-domain adaptation by clustering the development set and optimising the translation model mixture weights for each cluster. For a given test sentence, the nearest cluster is selected and its mixture weights used for combining the cluster-specific translation models.

2.3.4 Topic adaptation

Most of the work on topic adaptation for machine translation is based on LDA or variants of LSA, as introduced in Section 2.2. Topic adaptation is different from the approaches in Section 2.3.1 in that the goal is to automatically find structure in the training and test data that helps to differentiate between different possible translations. While the approaches in Section 2.3.1 make use of lexical or part-of-speech information, topic adaptation makes use of much lower-dimensional contextual information. Here we group the most relevant work according to the model type and the type of adapted features.

2.3.4.1 Bilingual topic modelling for word alignment and translation lexicon adaptation

The Bilingual Topic AdMixture Model (BiTAM) of Zhao and Xing (2006) is an LDA-style model to learn topic-dependent word alignments. The Hidden Markov Bilingual Topic AdMixture Model (Zhao and Xing, 2007) is an extension of BiTAM that integrates an HMM. According to the generative story, parallel documents are generated by first sampling a topic distribution for each document pair. For each sentence pair in the document, a topic is drawn from the document-level mixture and the words in the source sentences are generated according to a monolingual topic model. The target sentence is generated by first sampling an alignment link from a first-oder Markov process for each source position and then sampling foreign words at the aligned positions according to a topic-specific translation lexicon. These translation lexica are used to score phrase pairs depending on the topic mixture of a test document and yield BLEU

improvements of 0.41 over a GALE Chinese-English system with four reference translations. In short, these models are a combination of a monolingual source-side topic model and a bilingual word-based topic model.

Tam and Schultz (2007) propose a method for bilingual LSA-based translation lexicon adaptation. They train a Chinese LSA model, then bootstrap an English LSA model by initializing with the learned Dirichlet posteriors for Chinese which enforces a one-to-one topic correspondence between the source and target language. Language model and translation lexicon are adaptated by marginal adaptation. The overall improvement in terms of BLEU is small: 0.1 for translation lexicon adaptation, 0.2 for language model adaptation and 0.4 for the combination of both. Tam et al. (2008) employ the same technique to adapt the lexical weights of a translation system but report better results on a system trained on GALE Chinese-English data. BLEU improvements with adapted lexical weights are between 0.46 and 0.53, using four reference translations. Though the learning phase of the topic model is bilingual, it consists of two monolingual models with aligned topics.

2.3.4.2 SMT with monolingual topic models

Gong et al. (2010) build a monolingual source language topic model and compute for each entry in the phrase table the average topic distribution, given all training documents that contained the phrase pair. They group test sentences by topics and filter the phrase table according to a comparison of the maximum topic of each phrase pair and the test document. Gong and Zhou (2011) use the topical relevance of a target phrase and the maximum topic of a test sentence, computed using a mapping between source and target side topics, as an additional feature in decoding. The translation model is trained on a general-domain Chinese-English corpus (FBIS) plus a small corpus with text from five different domains (transport, sports, business etc.). Topic models are trained on the source and target sides of the domain-specific corpus. Gong et al. (2011) use a cache of topic words to reward target phrases with matching words during decoding. The topic cache is built up by extracting words from training documents that are similar to the test document. They experiment with Chinese to English translation using the FBIS corpus as training data and NIST development and test data. Combinations of several different caches yield BLEU improvements of up to 0.81.

Axelrod et al. (2012) build topic-specific translation models from the Chinese-English TED corpus and select additional topic-relevant data from the UN corpus to improve coverage. They split the dev and test sets into four topics and translate each

portion with the topic-specific translation model. Su et al. (2012) perform phrase table adaptation for Chinese to English translation in a setting where only monolingual in-domain data and parallel out-of-domain data are available. They train topic models separately on in-domain data (weblog text) and out-of-domain data (FBIS corpus) of the source language and use mappings from in-domain to out-of-domain topics for adaptation. Eidelman et al. (2012) use topic-dependent lexical weights as features in the translation model and achieve their best results with 10 topics and two Chinese to English systems trained on the FBIS corpus and Chinese Hansards.

2.3.4.3 SMT with monolingual topic models and similarity measures

Costa-jussà and Banchs (2010) build a vector space model that captures the source context of every training sentence that a phrase pair occurred in. Given a test input sentence and an applicable phrase pair, they compare the vector space representation of the test context to the vector space representation of all training instances for this phrase pair. A similarity feature enables the decoder to give priority to phrase pairs extracted from similar contexts[10]. Banchs and Costa-jussà (2011) extend this work by replacing the vector space representations with latent representations learned with Latent Semantic Indexing (LSI). The final score is the maximum of the similarity scores computed for the test context and every occurrence of the phrase pair in the training data. Similarly, Hewavitharana et al. (2013) perform dynamic adaptation with monolingual topic models for an English-Iraqi Arabic dialogue translation task. They encode topic similarity between a test conversation and all applicable training conversations in an additional feature.

Xiao et al. (2012) define a topic similarity model for a Chinese-English hierarchical phrase-based system trained on the FBIS corpus. They learn source and target side document topics and compute rule topic distributions from the document topic distributions. Correspondences between source and target side topics are learned using word-aligned training data and used to project source-side rule topics to target side rule topics. Four additional rule features are defined using the source-side and mapped target-side topic distributions: two features that measure the similarity of the test document topics and the rule topics and two features measuring rule sensitivity. The sensitivity of a rule is defined as the entropy of its topic distribution, with the intuition that rules with high-entropy topic distributions are less topic-specific and thus

[10]Their model is a phrase-based Spanish-English system trained on the Bible corpus (Chew et al., 2006).

more generally applicable.

Cui et al. (2014) extend the work of Xiao et al. (2012) by replacing topic models with neural networks. Source and target sentence representations are learned using denoising auto-encoders applied to bag-of-word representations[11]. The bag-of-word representations are expanded with retrieved documents from monolingual data sources by using the original sentences as queries. In a fine-tuning step, the source and target representations are made more similar by exploiting the sentence alignment signal. The source and target rule representations are averaged over the sentence representations and the same features are built as in Xiao et al.'s work, yielding improved performance over the LDA-based model.

2.3.4.4 SMT with word sense induction

Xiong and Zhang (2014) combine word sense disambiguation (WSD) and word sense induction (WSI) to induce a notion of word senses into the translation model. Word sense induction typically employs non-parametric methods to learn the latent sense of word types (Lau et al., 2010; Yao and Durme, 2011). In a multi-step approach, they first learn word sense clusters for source word types using the Hierarchical Dirichlet Process (HDP) and tag all words in the training, development and test data with their most likely senses given a small window of words around the source word token. Next, they train maximum entropy classifiers for all source word types to predict a distribution over target phrases using lexical and sense features in a window around the source word token. Possible disadvantages of the approach are that it involves hard sense tagging that discards the sense distributions and requires training a large number of models for both the WSI and the classification step. In addition, the fact that each WSI model learns word senses in its own low-dimensional space introduces sparsity into the classifiers as the number of senses grows with the number of source word types.

2.3.5 Discourse and document-level translation

Another line of research that is related to context-dependent translation deals with discourse phenomena in machine translation. For example, Carpuat (2009) investigate translation consistency within the same document as the "One translation per discourse" hypothesis for a French-English translation task. An oracle experiment where

[11] The input vectors are of the size of the vocabulary with all dimensions set to 0 except for the words that are part of the phrase for which they are set to 1.

document-level translation consistency is encouraged or forced to match the reference translations suggests that document consistency can benefit translation quality.

Foster et al. (2010b) model within-document structure by adapting the language model to document structure along the dimensions of *session*, *source language*, *speaker*, *title* and *section* which are annotated in the Hansard corpus, a corpus of Canadian parliamentary proceedings. They report modest improvements for both translation directions between French and English, with the best results produced by a linear combination of feature-specific models.

Tiedemann (2010) investigate context adaptation without specifically adapting the model to a domain or topic. Rather, they seek to address *repetition* and *consistency* of translations within the same document and use cache models to encourage consistent translations, where a decay factor accounts for the recency of cached items. They test both a unigram language model cache and a translation model cache which is filled with translation options from 1-best hypotheses. While the approach yields modest BLEU improvements on most test documents, one problem is that it relies on initial correct translation that can be promoted throughout the remaining document.

Gong et al. (2011) extend the work by Tiedemann (2010) and address the issue of incorrect initial translations. In addition to a dynamic translation model cache that stores phrase pairs of recently translated sentences in the same document, they introduce a static cache and a topic cache. Using a set of similar document pairs from the training data for each test document, the static cache is filled with phrase pairs from similar documents while the topic cache is filled with topical target words according to a monolingual topic model on the target side. Both additional caches help to improve the quality of the initial translations, while the dynamic cache helps to promote document consistency.

Louis and Webber (2014) further extend this line of work by introducing document structure to the topic cache. The system resets the cache at either predefined or automatically induced paragraph boundaries in biography documents, yielding modest incremental improvements over a baseline system that was not trained on biographies.

Hardmeier and Nivre (2012) introduce document-wide decoding as way of dealing with document-wide dependencies during translation, such as discourse phenomena. Based on an initial state representing a fully translated document, the decoder can make local changes, picking one sentence at a time in order to improve scoring functions that can be defined over the entire document. Permissible operations are changing phrase translations, changing the order of translated phrases and changing the source segmen-

tation. A case study on *lexical cohesion* using a semantic n-gram model over content words demonstrates the feasibility of the approach as well as small improvements in BLEU and NIST scores.

2.3.6 Cross-lingual semantic similarity for SMT

In this section, we describe methods that do not perform adaptation but try to improve the mapping between source and target language by introducing semantic information. Chen et al. (2010) employ a vector space model to represent the semantics of phrases. Following the *distributional hypothesis* (Harris, 1954), the source and target sides of translation rules in a hierarchical phrase-based model are represented by their source and target context vectors respectively, where the context consist of words from the initial phrase pair a rule was extracted from. The semantic similarities between source and target rule sides are used as decoding features.

A similar decoding feature is used by Zou et al. (2012) who train word embeddings in a bilingual fashion in order to make the embeddings comparable between source and target language. They use average word embeddings to represent phrases and measure cross-lingual similarity using the cosine function.

Similar to the ideas in Chen et al. (2010), Gao et al. (2013) learn source and target phrase representations with a multi-layer neural network that takes bag-of-words representations of phrases as input. They use the dot product between source and target representations as a semantic similarity feature. An interesting comparison of different model variants shows that learning phrase translations directly is more effective than decomposing phrases into words and computing word-word similarities as in lexical weighting schemes (Koehn et al., 2003).

2.4 Conclusion

In this chapter, we have introduced important concepts in machine translation and topic modelling and reviewed the relevant related work for the following chapters. We have categorised different lines of work related to contextual adaptation and in particular reviewed past approaches to domain adaptation and topic adaptation.

We have also discussed the related areas of discourse-aware translation and approaches using cross-lingual semantic similarity. While work on discourse in MT shares some of our assumptions about useful levels of structure for translation, the

goals differ in that most of these methods aim for consistency and coherence rather than optimal adaptation to a context. The goal of the approaches using cross-lingual similarity is to improve translational equivalence, but is related to our proposed work by the common assumption that explicitly capturing semantic information of phrases can improve translation.

The related work on dynamic topic adaptation, which is the starting point for our work, can be improved upon in a number of ways. A lot of the previous work uses monolingual topic models while we explore a bilingual topic model in order to better capture the meaning clusters that are important to translate between two languages. Previous work employing bilingual topic models (Zhao and Xing, 2006, 2007; Tam and Schultz, 2007; Tam et al., 2008) was limited to learning translation lexica and using them to score phrase pairs. The only previous work that directly adapts the phrase translation probabilities (Su et al., 2012) derives topic-specific phrase probabilities either from word-topics or from sentence-topics and targets cross-domain adaptation to a monolingual in-domain corpus. Instead, part of our work aims to directly estimate topic-dependent phrase translation probabilities for dynamic adaptation. Different from previous work where topics are learned at the sentence or document level, we will also explore the option of directly learning topic distributions for translation units to capture their distributional properties.

So far, the literature on domain and topic adaptation methods has developed largely in parallel and the approaches have not been compared to each other. We aim to close this gap by directly comparing our approaches to domain-adapted systems and also attempt to combine domain and topic adaptation approaches.

While data selection methods for translation and language model adaptation have shown to yield good results, we do not include them as baseline systems in our evaluations but favour other domain adaptation approaches instead. This choice is mostly based on the rather small-scale experimental setup in this thesis. Data selection methods are typically applied in situations where a large amount of training data is available from which (pseudo) in-domain data is selected prior to training the translation system.

Chapter 3

Adaptation with Sparse Features and Discriminative Training

In this chapter, we are concerned with the task of translating TED talks, which are transcribed speeches from recordings at the TED conference[1]. The TED corpus is an interesting data set because it has the characteristics of a domain as defined in Section 2.3 in that all contained documents originate from spoken language. On the other hand, the talks cover a variety of topics which makes the corpus more diverse in that respect than other corpora, such as the Europarl or News Commentary corpus.

We present an approach to *domain adaptation* for SMT that enriches standard phrase-based models with lexicalised word pair and phrase pair features to help the translation model select appropriate translations for sentences in TED talks. Our focus is on biasing a standard translation system for the vocabulary and style of the target domain. In addition, we demonstrate an approach to *topic adaptation* by incorporating source-side sentence-level topics to make the sparse features differentiate between more fine-grained topics within the TED domain.

We explore and compare several discriminative training approaches to include *sparse features* into translation systems trained under different data conditions. The idea is that sparse features can be added on top of baseline systems that are trained in the usual fashion, overlapping with existing features in the phrase table. This gives us flexibility to explore new feature sets which is particularly useful for training large systems from mixed-domain data. Specifically, we compare tuning on a small development set to tuning on an entire parallel in-domain corpus and introduce a new method of porting trained features to larger mixed-domain models.

[1] www.ted.com

Experimental results on data provided for the IWSLT 2012 shared task show that sparse lexicalised features can improve performance over a baseline with only dense features and that in some cases we get additional improvements with topic-specific features. We evaluate our methods on English-French and German-English.

3.1 Sparse features as model extensions

Sparse features such as word and n-gram indicator features have been succesfully used in many NLP systems in the past, for example for part-of-speech tagging (Collins, 2002) and parsing (McDonald et al., 2005). Similar methods have subsequently been applied to machine translation (Liang et al., 2006; Watanabe et al., 2007; Chiang et al., 2008, 2009), however they have not been adopted as quickly by the research community and even today only few groups have shown successful application of sparse features that significantly improve translation quality.

The ideas behind sparse features are quite simple. They provide the ability to design potentially overlapping feature sets that do not need to have a probabilistic foundation. This makes it easy to capture desirable properties of the system output and score hypotheses with a multitude of features that each capture some property of the output. In statistical machine translation, a similar set of dense features has been in use for about a decade. Sparse features can be used to extend the existing feature set and express additional preferences for the output. In principle, there are no restrictions on the type of features and they can also overlap the feature space that is already covered by dense features. In this chapter, we show how lexical selection can be influenced with sparse features that overlap with the dense phrase table features.

3.2 Tuning small and large feature sets

The traditional features used in SMT systems, such as phrase table features, language model features and lexicalised reordering features, are referred to as *dense* features because they are active for each sentence under translation, while sparse features are active in much fewer cases, depending on their level of generality. The dense features are usually trained in two steps. First, they are trained outside of the translation system to optimise the log-likelihood of the training data. In a second step, they are integrated into the translation system which is tuned for translation quality as measured by BLEU. This second step is important because the BLEU score has been shown to be correlated

with human judgements on translation quality (Papineni et al., 2002). This makes it a better training objective than log-likelihood, also because it is usually more effective to use the same criterion for training and evaluation. For a comparison of training objectives see Och (2003).

Up to recently, the standard procedure for tuning MT features was to run Minimum Error Rate Training (MERT) (Och, 2003), which is a discriminative training method that employs line search to optimise each feature dimension in turn. While it has proven to be very effective for small sets of 10-20 features, its sequential nature causes problems when applied to large feature sets. Finding an optimum in a high-dimensional space is very difficult when only one dimension at a time can be explored. In order to overcome this problem, several different online or batch algorithms have been proposed that can in theory deal with an arbitrary number of features. Examples are online MIRA (Liang et al., 2006; Watanabe et al., 2007; Chiang et al., 2008, 2009; Chiang, 2012), Pairwise Ranked Optimisation (PRO) (Hopkins and May, 2011) and batch MIRA (Cherry and Foster, 2012). While for online MIRA, decoding and weight updates are performed for a single development sentence or a mini-batch, PRO and batch MIRA first decode the entire development set with the current weight set before doing an optimisation run based on all examples. This is more efficient in practice and has been shown to perform comparably or better than fully online learning (Cherry and Foster, 2012).

Chiang (2012) extended previous work on online learning by introducing Adaptive Regularisation of Weights (AROW) for MT tuning. AROW adapts the learning rate of the weight update for each feature dimension depending on the amount and size of previous feature updates. This is useful to allow for larger updates for features that occur rarely (sparse features). Dense features get updated frequently and therefore the learner should much sooner have confidence in the learned weights and decrease their learning rates for the remainder of the training phase.

A further extension to online learning has been proposed by Green et al. (2013) who use an update rule based on AdaGrad. The idea is similar to AROW but while AROW includes both adaptivity and conservativity[2], AdaGrad only provides adaptivity. Green et al. report that preliminary results showed adaptivity to be more important than conservativity and AdaGrad to be more robust.

[2]While for MIRA, conservativity is ensured by including the squared norm of the old and new weights vectors into the objective function, AROW includes a KL-divergence term between two Gaussians modelling the old and new weight vector.

3.3 Related work

We have reviewed previous work on domain adaptation in Section 2.3.3. Of that work, the most relevant for this chapter is by Su et al. (2012) who employ Hidden Topic Markov Models (HTMMs) (Gruber et al., 2007) to train source-side topic models to adapt an out-of-domain table with monolingual in-domain data. We do not directly adapt any of the features in the phrase table but instead add overlapping sparse features, both with and without topic information from HTMMs.

MIRA has been proposed for tuning machine translation systems with large features sets, for example by Watanabe et al. (2007) and Chiang et al. (2009). Recent work that compares tuning on a small development set versus tuning on the entire training data has been presented in Simianer et al. (2012), though not with a focus on adaptation. The idea of using source triggers to condition word translation is somewhat related to the trigger-based lexicon models of Mauser et al. (2009), but they use context words as additional triggers and use them as extensions to the IBM word lexicon models or within classifiers trained for each target word.

3.4 Training sparse features for domain adaptation

Adding sparse, lexicalised features to existing translation systems is one way to bias the systems towards translating a particular domain, in our case the TED domain. We distinguish two data conditions for the baseline models: an in-domain model trained only on in-domain TED data and a mixed-domain model trained on TED data and large amounts of out-of-domain data. Our features are trained with the MIRA algorithm which is explained briefly in the following subsection. We compare the standard approach, e.g. tuning on a rather small development set, to the less common jackknife approach, details of which are given in subsection 3.4.3.

3.4.1 Training features with MIRA

Recently, the Margin Infused Relaxed Algorithm (MIRA) (Crammer and Singer, 2003) has gained popularity as an alternative training method to Minimum Error Rate Training (MERT) (Och, 2003), because it can deal with an arbitrary number of features. MIRA is an online large margin algorithm that enforces a margin between good and bad translations of the same sentence. This margin can be tied to a loss function like BLEU (Papineni et al., 2002) or another quality measure. Given that we can provide

the learning algorithm with good oracle translations, the model learns to score hypothesis translations with higher BLEU scores better than translations with lower BLEU scores. MIRA updates the feature weights of a translation model by iterating through the training data, decoding one sentence at a time and performing weight updates for pairs of good and bad translation examples.

We use a slightly modified version of the implementation described in Hasler et al. (2011) that selects hope and fear translations from a 30-best list instead of running the decoder with hope and fear objectives. This has the effect that there is no need for dynamically computed sentence-level BLEU scores anymore because real sentence-level BLEU scores can be computed on the 30-best list. Chiang (2012) mentions that certain features, e.g. the language model, are very sensitive to larger weight changes and so we introduce a separate learning rate for core features (translation model, language model, word penalty et cetera) in order to reduce fluctuations and keep MIRA training more stable. This learning rate is independent of the C parameter in the objective function solved by MIRA and is set to 0.1 for core features and 1.0 for sparse features.

3.4.2 Feature sets

We experiment with two classes of indicator features, sparse phrase pair features and sparse word pair (word translation) features. Word pair features capture translations of single source words to single target words, whereas phrase pair features capture translations of several words on the source side into several words on the target side. The class of phrase pair features depends on the decoder segmentation and can also include phrase pairs of length 1 on each side if such a phrase pair was extracted from the training data. Word pair features on the other hand depend on word alignment information and only contain word pairs that were connected by an alignment point in the training data.

Both of these feature classes are then extended with topic information acquired from topic models trained on the source side of the training corpus. The topic information is integrated as a source side trigger for a particular word or phrase pair, given a topic. Details about how these topic models were trained are given in section 3.5. Table 3.1 shows a pair of source sentence and hypothesis translation taken from a MIRA training run and examples of the features extracted from that sentence pair. The feature values indicate the number of times a feature occurred in a given sentence pair. The features in the first column capture general word or phrase translations while

Input (topic 10)	[a language] [is a] [flash of] [the human spirit] [.]
Hypothesis	[une langue] [est une] [flash de] [l' esprit humain] [.]
Reference	une langue est une étincelle de l' esprit humain .

General Features	**Topic-specific Features**
wp_a∼une=2	wp_10_a∼une=2
wp_language∼langue=1	wp_10_language∼langue=1
wp_is∼est=1	wp_10_is∼est=1
wp_flash∼ flash=1	wp_10_flash∼ flash=1
wp_of∼de=1	wp_10_of∼de=1
...	...
pp_a,language∼une,langue=1	pp_10_a,language∼une,langue=1
pp_is,a∼est,une=1	pp_10_is,a∼est,une=1
pp_flash,of∼flash,de=1	pp_10_flash,of∼flash,de=1
...	...

Table 3.1: Example English-French sentence pair with source segmentation (top) and the word pair (wp) and phrase pair (pp) features extracted from it.

the features in the second column capture translations given a particular topic (here: topic 10). The features without topic information simply indicate whether a particular word or phrase translation should be favoured[3] or avoided by the decoder, depending on whether they receive positive or negative weights during training. The features with topic information are triggered by the topic of the source sentence, that is, for a particular source sentence to be translated, only the features that were seen with the topic of that sentence will fire.

The TED domain is an interesting domain to try out these classes of features, because we can distinguish two different adaptation tasks: (1) adapting to the general vocabulary of TED talks as opposed to the vocabulary of out-of-domain texts (details in the experiments section), and (2) adapting to the vocabulary of subsets of TED talks that can be grouped into more fine-grained topics which we try to capture with topic models.

[3] in the domain represented by the development set

3.4.3 Jackknife training

Training sparse features always involves a risk of overfitting on the tuning set, especially with highly lexicalized features that might occur only once in the tuning set. Therefore, training sparse features on the entire training set used to estimate the phrase table is expected to be more reliable. For discriminative training methods this means that the training set needs to be translated in order to infer feature values and compute BLEU scores. However, translating the same data that was used to train the translation system would obviously cause overfitting as well, thus the system needs to be adjusted to prevent this. In order to translate the whole training data without bias, we apply the jackknife method to split up the training data into n folds (here: $n = 10$). We create n subsets of the training data containing $n-1$ folds and leaving out one fold at a time. These subsets serve as training data for n translation systems that can be used to translate the respective held out fold.

To use the jackknife systems for MIRA training, we modified the algorithm to accept n sets of decoder configuration files, input files and reference files. Instead of running n instances of the same translation system in parallel, we run n jackknife systems in parallel and average their weight vectors several times per epoch.

When applying the jackknife method to the TED in-domain data, we noticed a problem with this approach. Usually it would be good practice to create folds in a way that the resulting subsets of training data are as uniform as possible in terms of vocabulary to minimize the performance hit caused by the missing fold. We did this by simply splitting the training data in way that the fold of a sentence would be determined by the result of "$line\_nr$ mod n". However, the vocabulary of the TED data turned out to be quite repetitive across sentences belonging to the same talks. Thus, splitting the data in this way had the effect that each of the n systems had a certain amount of overlap between training and heldout data. This resulted in a preference for longer phrases, overly long translations on the test set and decreasing performance during MIRA training.

We were able to overcome the overfitting effect of line-wise data splits by splitting the data in a block-wise fashion instead. That is, the first $corpus\_size/n$ lines were assigned to fold 1, the following block to fold 2 and so on. This way the training data was much less likely to overlap with the held out fold. The results on a held out set during MIRA training (in particular the length penalty and overall length ratio) showed that this helped to prevent overfitting on the held out fold.

3.4.4 Retuning features for mixed-domain models

Tuning sparse features on top of large translation models can be time and memory-consuming. Especially the jackknife approach causes immense overhead to tune with the mixed-domain data because we would need to train n different phrase tables that all include most of the in-domain data and all of the out-of-domain data[4]. In addition, each change to the mixed-domain models would require repeating the jackknife tuning, while the retuning method is faster and more feasible to repeat multiple times.

Therefore, we wanted to investigate whether there is an alternative way of tuning the sparse features on all of the in-domain data while also making use of the out-of-domain data. Tuning with the in-domain models allows for more flexibility in the training setup because the data set is relatively small. Since our goal is to translate TED talks, we assume that tuning sparse features only on the TED corpus should provide the model with enough information to select the appropriate vocabulary. Therefore, we propose to port the features that were tuned on the in-domain model over to the mixed-domain model. The advantage of this method is that features can be tuned on all of the in-domain training data (jackknife tuning) or in other ways that are feasible on a smaller in-domain model but might not scale well on a large mixed-domain model.

However, porting tuned feature weights from one model to another is not straightforward because the scaling of the core features is likely to be different. Therefore, to bring the sparse feature weights on the right scale to integrate them into the mixed-domain model, we perform a retuning step with MIRA. We take the sparse features tuned with the jackknife method and combine them into a single aggregated meta-feature with a separate, global weight. During decoding, the meta-feature weight is applied to all sparse word or phrase pair feature weights. In the retuning step, the core weights of the mixed-domain model are tuned together with the meta-feature weight.

An overview of our tuning schemes is given in Figure 3.1. The training step denotes the entire training pipeline yielding the baseline models. Direct tuning refers to tuning with MIRA on a small development set and applies to both kinds of baseline models, while *jackknife tuning* only applies to in-domain models and *retuning* only to mixed-domain models.

[4]Training the mixed-domain system for the English-French language pair took more than a week.

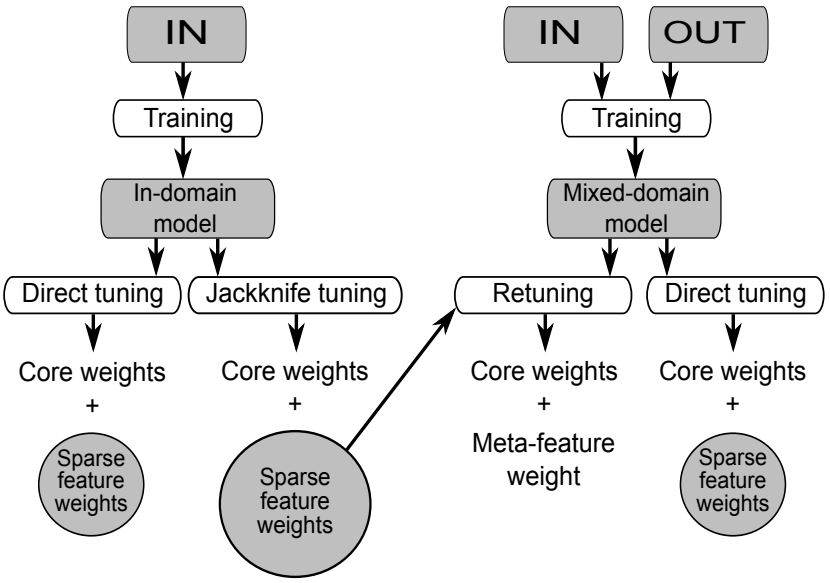


Figure 3.1: In-domain (IN) and mixed-domain (IN+OUT) models with three tuning schemes for tuning sparse feature weights: direct tuning, jackknife tuning and retuning.

3.5 Training topic models

The topic models used for building topic-dependent word pair and phrase pair features are Hidden Topic Markov Models (HTMM) (Gruber et al., 2007) and were trained with a freely available toolkit. While topic modelling approaches like Latent Dirichlet Allocation assume that each word in a text was generated by a hidden topic and the topics of all words are assumed to be independent, HTMMs model the topics of words in a document as a Markov chain where all words in a sentence are assigned the same topic. This makes intuitively more sense than assigning several different topics within the same sentence and Gruber et al. (2007) show that HTMMs also yield lower model perplexity than LDA. The former characteristic makes HTMMs particularly suitable for our purpose. We are guaranteed that each word in a source phrase is assigned the same topic and therefore we do not have to figure out how to assign phrase topics given word topics.

HTMMs compute the joint conditional probability of the latent variables $P(z_n, \psi_n | d, w_{i=1}, .., w_{N_d})$ for each sentence, where z_n is the topic of sentence n and ψ_n determines the topic transition between words and can be non-zero only at sentence boundaries. d is the document and w_i are words in the document d. When $\psi_n = 0$, the topic is identical to the previous topic, when $\psi_n = 1$, a new topic is drawn from a

distribution θ_d. Once a sentence topic has been selected, all words w_i are generated according to a multinomial distribution with topic-specific parameters. In order to assign topics to sentences in our training data, we derive a sentence topic distribution

$$\begin{aligned} P(topic|sentence) &= P(z_{d,n}|d, w_{i=1}, .., w_{N_d}) \\ &= P(z_{d,n}, \psi_{d,n} = 0|d, w_{i=1}, .., w_{N_d}) \\ &+ P(z_{d,n}, \psi_{d,n} = 1|d, w_{i=1}, .., w_{N_d}) \end{aligned} \tag{3.1}$$

We noticed that the distributions $P(topic|sentence)$ were quite peaked in most cases and therefore we tried to use a more compact representation. First, we selected the most likely topic according to the topic distribution and treated this as ground truth, ignoring all other possible topics. Alternatively, we selected the two most likely topics along with their probabilities, ignoring the second most likely topics with a probability lower than 30%. The topic probabilies were then used instead of the binary feature values in order to integrate the confidence of the topic model in its assigments. Experimental results were slightly better for the maximum-topic representation without probabilities and therefore we chose this simpler representation in all reported experiments.

In order to improve the quality of the topic models, we used stop word lists and lists of salient TED talk terms to clean the in-domain data before training the topic models. All TED talks come with a small set of keywords ($\sim$300 in total) describing the content of the talk. The idea was to use the information contained in these keywords to select salient terms that frequently cooccur with the keywords. We first computed TF-IDF scores for all words in each talk, normalised by the number of words in the talk to make them comparable across documents. We then summed up the normalised TF-IDF scores for each keyword, i.e. the scores of words in all documents associated with a particular keyword, and selected the top 100 terms for each keyword. This yielded $\sim$10500 terms for English and $\sim$11700 terms for German.

In cases where this filtering yielded empty sentences in the in-domain data (sentences with no salient terms), the topic label was set to *unk*. We ran the topic training for 100 iterations and trained 30 topics over training, development and test sets. We modified the Moses decoder to accept topic information as XML mark-up and annotated all data with sentence-wise topics (and optionally the respective probabilities). Table 3.2 gives some examples of topics and their 5 most frequent terms for English and German as a source language, as we use topic triggers associated with the source sentence for our sparse features. The topic models represent topics as integers but here

cancer (topic 3)	***ocean*** (topic 9)	***body*** (topic 25)	***universe*** (topic 29)
cancer	water	brain	universe
cells	ice	human	space
body	surface	neurons	Earth
heart	Earth	system	light
blood	Mars	mind	stars
disease	ocean	cells	matter
cell	feet	brains	black
cancer (topic 17)	***ocean*** (topic 20)	***body*** (topic 25)	***universe*** (topic 9)
Krebs	Wasser	DNA	Erde
Patienten	Meer	Leben	Universum
Gehirn	Menschen	Licht	Planeten
Zellen	Ozean	Bakterien	Leben
Körper	Tiere	Menschen	Sonne
Herz	Welt	System	Milliarden
Jahre	Fisch	Zelle	Raum

Table 3.2: Sample English (top) and German (bottom) HTMM topics along with their manual labels and topic ids.

we have added labels to indicate the nature of the topics and we selected topics that map across the two languages. In general, the topics do not map to equivalent topics in another language.

Figure 3.2 shows a sequence of training sentences and their most likely topic (as well as the second most likely topic if applicable). We can see that for some of the sentences, the model assigns what we have labelled the *universe* topic with high probability while for others it is less certain or transitions to the *ocean* topic.

3.6 Experimental setup

We evaluate our training schemes on English-French and German-English translation systems trained on the data sets as advised for the IWSLT 2012 TED task. As in-domain data we used the TED talks from the WIT^3 website[5] (Cettolo et al., 2012). As out-of-domain data we used the Europarl, News Commentary and MultiUN (Eisele and Chen, 2010) corpora and for En-Fr also the 10^9 corpus taken from the WMT2012

[5]https://wit3.fbk.eu/mt.php?release=2012-03

universe (0.41)	"And physicists came and started using it sometime in the 1980s."
universe (0.47)	"And the miners in the early part of the last century worked, literally, in candle-light."
ocean (0.71)	"And today, you would see this inside the mine, half a mile underground."
ocean/universe (0.51/0.49)	"This is one of the largest underground labs in the world."
universe (0.99)	"And, among other things, they're looking for dark matter."
universe (1.00)	"There is another way to search for dark matter, which is indirectly."
universe (1.00)	"If dark matter exists in our universe, in our galaxy, then these particles should be smashing together..."

Figure 3.2: Topic assignment to training sentences with topic probabilities in brackets.

release. An overview of all training data as well as development and test data is given in Table 3.3.

With this data we trained in-domain and mixed-domain baselines for both language pairs. For the mixed-domain baselines (trained on data from all domains), we used simple concatenations of all parallel training data, but trained separate language models for each domain and linearly interpolated them on the development set. All systems are phrase-based systems trained with the Moses toolkit (Koehn et al., 2007). Compound splitting and syntactic pre-reordering was applied to all German data. As optimisers we used MERT as implemented in the current version of Moses and a modified version of the MIRA implementation in Moses as described in Section 3.4.1. We provide baseline results for tuning with both MERT and MIRA for comparison, though our model extensions are evaluated with respect to the MIRA baselines. Reported BLEU scores were computed using the mteval-v11b.pl[6] script.

All experiments except the jackknife experiments used the TED dev2010 set as development set (dev). The TED test2010 set was split into two parts, test2010.part1 and test2010.part2. For the in-domain experiments, one part was used to select the

[6] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl

Data set	En-Fr	De-En
TED talks	140K (1029 talks)	130K (976 talks)
Europarl v7	2M	1.9M
News Commentary v7	137K	159K
MultiUN	12.9M	161K
10^9 corpus	22.5M	n/a
total	35.9M	2.3M
TED talks (monolingual)	143K	142K
dev2010	934 (8 talks)	900 (8 talks)
test2010.part1	898 (5 talks)	665 (5 talks)
test2010.part2	766 (6 talks)	900 (6 talks)

Table 3.3: Number of sentences in in-domain (TED talks) and out-of-domain training data used in our systems.

best weights found during MIRA training and the other part was used for evaluation, respectively. We refer to these sets as test1 and test2 to indicate which of the two parts was used as the test set. We note that test1 and test2 yield quite different BLEU scores for the baseline models. However, Tables 3.5 and 3.6 show that the relative improvements achieved with MIRA are roughly proportional and thus we will report results on just one of the two sets for experiments on the mixed-domain baselines.

The size of the feature set was 5K-15K when training on a development set and 60K-600K when training on all training data, depending on the feature type.

All MIRA experiments were initialized with the tuned weights of the MERT baselines. MIRA experiments on the dev set were run for 20 epochs, retuning experiments for 10 epochs and jackknife experiments on the entire training set for 2 epochs. All reported MERT results were averaged over three runs while MIRA was run in a deterministic fashion.

3.7 Results

We are evaluating the impact of our sparse features on the in-domain and mixed-domain systems[7]. Tables 3.5 and 3.6 show the results on the in-domain system with

[7]The models presented here are comparable in performance to our models for the IWSLT 2012 evaluation campaign without the additional monolingual data from News Crawl and Gigaword. Our systems were ranked first for English-French and second for German-English (Hasler et al., 2012a).

		BLEU(test1)	BLEU(test2)
En-Fr	MERT(dev) IN	28.6 (0.969)	30.9 (0.963)
	MIRA(dev) IN	29.4 (0.987) (+)	31.7 (0.982) (+)
De-En	MERT(dev) IN	26.6 (0.987)	29.9 (1.001)
	MIRA(dev) IN	26.3 (0.955) (-)	29.6 (0.969) (-)

Table 3.4: Changes to the length ratio (hypotheses/reference, in brackets) between MERT and MIRA tuning, indicated by (+) and (-).

BLEU scores reported on both parts of the test2010 set, using the respective other part as devtest set. The largest improvements over the MIRA baseline are marked in bold and the relative changes are indicated in brackets. The sparse feature sets are added to the baseline systems separately, not cumulatively.

3.7.1 In-domain models

First we note that MIRA training improves the MERT baseline performance for the English-French system by 0.8 BLEU on both test sets, but decreases performance for the German-English system by 0.3 BLEU. We believe that this divergence has to do with the changes in length ratio after MIRA training, as shown in Table 3.4. For English-French, translations get longer during MIRA training while for German-English they get shorter, incurring an increased brevity penalty according to the BLEU score. This trend persists as we add sparse features to the models as described in the following.

Since MIRA has quite a different impact on the translation performance with the core features (translation model, reordering model, language model, word penalty, phrase penalty), we focus on the impact of sparse features with respect to the MIRA baselines.

English-French results In Table 3.5, we observe that all sparse feature setups beat the MERT baseline and most of them beat the MIRA baseline. For the MIRA experiments with features tuned on the dev set (top of the table), we notice that phrase pair features seem to perform better than word pair features on both test sets and sparse features with topic triggers seem to do better than sparse features without topic information.

The results of the MIRA experiments using the jackknife method are in most cases

En-Fr	BLEU(test1)	BLEU(test2)
MERT(dev) IN	28.6	30.9
MIRA(dev) IN	29.4	31.7
MIRA(dev)		
+ wp	29.2 (-0.2)	31.6 (-0.1)
+ wp_topics	29.5 (+0.1)	31.8 (+0.1)
+ pp	**29.6** (+0.2)	31.7 (+0.0)
+ pp_topics	**29.6** (+0.2)	**31.9** (+0.2)
MIRA(jackknife)		
+ wp	29.7 (+0.3)	**32.2** (+0.5)
+ wp_topics	29.5 (+0.1)	32.1 (+0.4)
+ pp	**29.9** (+0.5)	**32.2** (+0.5)
+ pp_topics	29.6 (+0.2)	32.0 (+0.4)

Table 3.5: In-domain baselines (IN) and results for sparse feature training on En-Fr in-domain model, training on a development set (dev) and on all training data (jackknife).

better than the results trained on the small dev set. We get an increase of up to 1.3 BLEU over the MERT baseline and up to 0.5 BLEU over the MIRA baselines. This shows that the jackknife method is better suited to train sparse features than training on a small dev set. We still observe slightly better results for phrase pair features than for word pair features, even though this observation is less conclusive as compared to their improvements on the development set. However, the topic-dependent features do not improve over the simple features in this setup. This is a surprising result because we would expect that jackknife tuning is more effective at mitigating sparsity problems than tuning on a small development set.

German-English results In Table 3.6, we observe an overall slightly better performance with the word pair features. Here, the simple and topic-dependent features perform similarly, but the topic-dependent features outperform the simple ones only in one out of four cases. Again, we see better results with jackknife tuning than with tuning on the development set, with an increase of up to 0.2 BLEU over the MERT baseline and up to 0.7 BLEU over the MIRA baselines. However, as for the English-French system the simple sparse features perform better than their topic-dependent counterparts in this setup.

De-En	BLEU(test1)	BLEU(test2)
MERT(dev) IN	26.6	29.9
MIRA(dev) IN	26.3	29.6
MIRA(dev)		
+ wp	**26.7** (+0.4)	**29.8** (+0.2)
+ wp_topics	26.6 (+0.3)	29.7 (+0.1)
+ pp	26.5 (+0.2)	29.7 (+0.1)
+ pp_topics	26.4 (+0.1)	**29.8** (+0.2)
MIRA(jackknife)		
+ wp	**27.0** (+0.7)	**30.1** (+0.5)
+ wp_topics	26.4 (+0.1)	29.7 (+0.1)
+ pp	26.8 (+0.5)	30.0 (+0.4)
+ pp_topics	26.4 (+0.1)	29.8 (+0.2)

Table 3.6: In-domain baselines (IN) and results for sparse feature training on De-En in-domain model, training on a development set (dev) and on all training data (jackknife).

Feature set combinations For English-French, we also experimented with combinations of general sparse features and topic-specific features but observed a decrease in performance rather than an improvement. This may have to do with the fact that the feature sets are highly overlapping and there currently is no mechanism for backing off to general features when no topic-specific features are available[8]. Instead, if a topic-specific feature exists, then both the topic-specific and the general feature will fire. We also observed a decrease in performance when combining word pair and phrase features and therefore only used one of the feature sets in subsequent experiments. However, we only ran these experiments on the small development set, so it is possible that the effect we were observing was due to overfitting. Training these feature class combinations on the full in-domain data could help to answer this question.

3.7.2 Mixed-domain models

Tables 3.7a and 3.7b show results on the mixed-domain models, where we observe a similar divergence in performance between the MERT and MIRA baselines as on the in-domain models: a plus of 1.1 BLEU for English-French and a minus of 0.4 BLEU for German-English. The first block of results refers to MIRA training on the dev2010

[8]This would be possible to implement, though.

En-Fr	BLEU(test1)
MERT(dev) IN+OUT	30.0
MIRA(dev) IN+OUT	31.1
MIRA(dev), direct tuning	
+ wp	**31.6** (+0.5)
+ wp_topics	31.4 (+0.3)
+ pp	31.4 (+0.3)
+ pp_topics	31.5 (+0.4)
MIRA(dev), retuning	
+ wp	**31.6** (+0.5)
+ wp_topics	31.1 (+0.0)
+ pp	31.5 (+0.4)
+ pp_topics	31.3 (+0.2)

(a) English-French

De-En	BLEU(test1)
MERT(dev) IN+OUT	27.2
MIRA(dev) IN+OUT	26.8
MIRA(dev), direct tuning	
+ wp	**26.9** (+0.1)
+ wp_topics	**26.9** (+0.1)
+ pp	**26.9** (+0.1)
+ pp_topics	26.7 (-0.1)
MIRA(dev), retuning	
+ wp	27.1 (+0.3)
+ wp_topics	**27.2** (+0.4)
+ pp	27.0 (+0.2)
+ pp_topics	27.0 (+0.2)

(b) German-English

Table 3.7: Mixed-domain baselines (IN+OUT) and results for sparse feature training on En-Fr and De-En mixed-domain model: direct sparse feature tuning and retuning with MIRA using jackknife-trained features.

set as for the in-domain models (direct tuning), while the second block results from the retuning setup described in Section 3.4.4 (retuning).

The direct approach gains up to 0.5 BLEU for English-French and up to 0.1 BLEU for German-English over the MIRA baselines, retuning with MIRA and jackknife features gains up to 0.5 BLEU for English-French and up to 0.4 BLEU for German-English over the MIRA baselines. This is another indication that sparse features trained with the jackknife method can leverage information from the in-domain training data to help the model select appropriate words and phrases for the target domain. In some cases we can observe that topic features improve over simple features, but in more of the cases they perform weaker. In general, the results show that features trained only on in-domain models can help to improve performance of much larger mixed-domain models. While for the in-domain models the results on both language pairs are similar with respect to the MIRA baselines, the results on mixed-domain models are clearly better for English-French. This may be due to the richer morphology in German, leading to more different word forms and thus more sparsity in word pair and phrase pair features. We experimented with lemmatised word forms as well but did not perceive

any gains in translation quality.

3.7.3 Potential improvements to feature training

Judging from the results in the previous sections, we suspect that there are sparsity issues that need to be addressed in order to get more benefit from our features, either by generalising the feature sets or by making the training algorithm more sensitive to sparse features. For example, we could use word classes instead of surface forms to generalise the features to more words and phrases.

We attempted to account for the frequency difference between dense and sparse features by setting different learning rates and pretuning the dense features. However, this approach is quite coarse and requires special treatment of the feature classes. In addition, it does not deal with varying feature frequency within the sparse feature sets. Subsequent work by Green et al. (2013) propose a new algorithm for sparse feature training with an adaptive learning rate and employ similar feature sets to the ones described in this chapter[9]. They experiment with different sizes of tuning sets and find that while larger tuning sets improved performance, there is also an increased domain effect when the bitext domain does not match the test set domain. Since we aim for exactly this domain effect in tuning our features on an in-domain training corpus, we believe that we could improve the performance of the jackknife setup by including an adaptive learning rate as proposed by Green et al.

3.7.4 Qualitative evaluation of topic features

Distribution across data sets For the English-French in-domain systems trained on development data, we see an improvement of topic features over simple sparse features. That these effects are not stronger might be due to the quite diverging distributions of topics across training, dev, devtest and test sets. Figure 3.3 shows the number of sentences in dev, devtest and test data[10] labelled with each of the topics (0-29). For example, the *universe* topic (topic 29) appears quite frequently in the training and dev data, but only twice in test2 and never in test1. For future experiments with sentence-level topic features it should be ensured that topics are distributed more evenly across the data sets, so that improvements on topics found in the training and development

[9]Their model is a phrase-based system based on the alignment templates of Och and Ney (2004). The feature set includes discriminative rule indicators, alignment indicators and reordering features.

[10]Training data counts were between 2252 and 7170 sentences per topic.

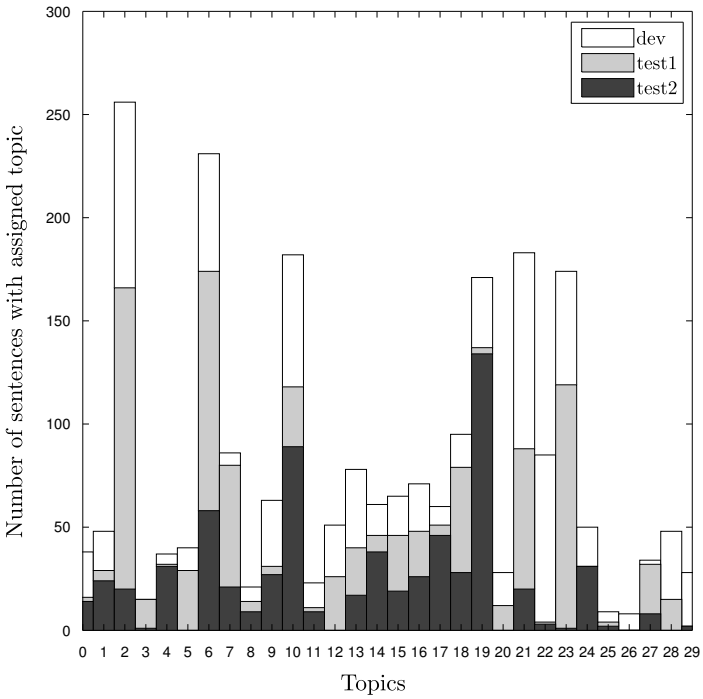


Figure 3.3: Distribution of topics in dev, test1, test2.

data can be measured on the test sets.

Tuned feature weights related to *matter* Lexicalised features with topic triggers are even sparser than simple lexicalised features and therefore we would expect that they benefit particularly from jackknife tuning. However, our current results show the opposite tendency in that topic features seem to do worse than simple features under the jackknife setup. Table 3.8 gives an example of word pair features trained with the jackknife method, with and without topic information. It shows the features with the largest positive/negative weights (those with the highest discriminative power learned by the model) related to the English source word *matter*. In the left table, we see that both models with word pair features have learned that *matière* is the most appropriate French translation for the English word *matter*. Both models penalise some translations of the other word sense like the French word *important*. However, the model without topic information considers *importe* an almost equally likely translation, while the model with topic information penalizes all translations that do not preserve the physical word sense (as in *dark matter*). As mentioned above, the *universe* topic did not appear at all in test1, so the impact of features related to this topic has not been measured in the evaluation.

In the right table, we see the learned feature weights for the phrase pair features

Sparse feature	Weight
wp_matter∼matière	0.00170
wp_matter∼importe	0.00107
wp_matter∼importent	0.00027
wp_matter∼important	-0.00037
wp_29_matter∼matière	0.00431
wp_29_matter∼importent	-0.00001
wp_29_matter∼importe	-0.00134
wp_29_matter∼important	-0.00172

Sparse feature	Weight
pp_matter∼la matière	0.01122
pp_matter∼importe	-0.00101
pp_matter∼matière	-0.00136
pp_matter∼important	-0.00162
pp_29_matter∼la matière	0.00512
pp_29_matter∼importe	-0.00027
pp_29_matter∼important	-0.00078
pp_29_matter∼matière	-0.00244

Table 3.8: Examples of En-Fr jackknife-trained word pair and phrase pair features, with and without topic information (topic 29: *universe*).

with and without topic information. Apart from discriminating the physical word sense of *matter* from its more general sense, the phrase pair features also discriminate between the translations *matière* and *la matière* and clearly prefer the latter over the former. This is because while the English phrase *dark matter* mostly occurs without determiner, its French translation *la matière noire* does require a determiner.

3.7.5 Qualitative evaluation of translation output

Figure 3.4 shows examples from the output of the English-French system with phrase pair features, Figure 3.5 for the German-English system with word pair features, both trained with the jackknife method. We note that for all of these examples, the addition of the sparse features seems to have an effect on lexical choice. For example, the output becomes more fluent in the first example with phrase pair features and a better noun translation is chosen in the second example. For the examples with word pair features, we also notice improved lexical selection in the translation of nouns.

Table 3.9 shows the learned sparse weights related to the second example for both language pairs. In both cases, the model has learned a preference for the correct translation. These results indicate that sparse word pair and phrase pair features can be used to bias lexical choice. However, a lot of care has to be taken to prevent overfitting that can result in models that do not generalise well to new test sets.

Source	The more money you earn the more *satisfied you are*.
Baseline	Le plus d'argent vous gagnez plus satisfaits vous.
+ pp	Le plus d'argent vous gagnez plus **vous êtes satisfait**.
Reference	Plus vous gagnez d'argent, plus **vous êtes satisfait**.
	Adversity is just *change* that we haven't adapted ourselves to yet.
Baseline	L'adversité est juste changer que nous n'avons pas encore adapté à nous-mêmes.
+ pp	L'adversité est juste **changement** que nous n'avons pas encore adapté à nous-mêmes.
Reference	L'adversité est juste un **changement** auquel nous ne nous sommes pas encore adaptés.

Figure 3.4: Example output from the English-French system with sparse phrase pair features, trained with the jackknife method (Table 3.5).

Source	Sie züchteten *Fleischrinder* auf dem, was im Grunde Feuchtgebiete waren.
Baseline	They raised meat cattle on what were basically wetlands.
+ wp	They raised **beef cattle** on what were basically wetlands.
Reference	They raised **beef cattle** on what was essentially wetlands.
Source	Der Mikroprozessor ist ein Wunder. Der PC ist ein *Wunder*.
Baseline	The microprocessor is a miracle. The PC is a wonder.
+ wp	The microprocessor is a miracle. The PC is a **miracle**.
Reference	The microprocessor is a miracle . The personal computer is a **miracle**.

Figure 3.5: Example output from the German-English system with sparse word pair features, trained with the jackknife method (Table 3.6).

Sparse feature	**Weight**
pp_change∼changer	-0.02199
pp_change∼changement	0.01061

Sparse feature	**Weight**
wt_Wunder∼wonder	-0.00156
wt_Wunder∼miracle	0.00096

Table 3.9: Learned feature weights for examples in Figure 3.4 and Figure 3.5.

3.8 Conclusion

We have presented a novel way of training lexicalised features for a domain adaptation setting by adding sparse word pair and phrase pair features to in-domain and mixed-domain models. In addition, we proposed a method for using topic information

derived from Hidden Topic Markov Models trained on the source language to condition the translation of words or phrases on the sentence topic. This was shown to yield improvements over simple sparse features on English-French in-domain models, while for the German-English models the simple features without topic information yielded better results in most cases. We experimented with the jackknife method to use the entire in-domain data for feature training and showed BLEU score improvements for both language pairs over tuning on a small development set. However, we did not observe improvements with the topic-dependent sparse features when trained with this method, probably due to the increased sparsity of topic-dependent features and the lack of explicit backoff features. Finally, we introduced a retuning method for mixed-domain models that allows us to adapt features trained on the entire in-domain data to the much larger mixed-domain models. This yielded improvements over the mixed-domain baselines of up to 0.5 BLEU for English-French and up to 0.4 BLEU for German-English.

The methods presented in this chapter offer new ways of adapting a given translation system to a specific target domain, as exemplified for the domain of TED talks. We saw moderate improvements for domain adaptation which, as mentioned in Section 3.7.3, could potentially be increased with a more robust learning algorithm. It may also be useful to include feature selection in order to reduce the noise introduced by adding a large number of features to the model. Finally, it is possible that the sets of sparse features explored here are too limited in their ability to generalise to unseen data and should be complemented with a set of more general (sparse) features.

We further experimented with topic adaptation using sparse features. However, while some of the experiments yielded small positive results, it seems that the increase in feature sparsity poses problems for topic adaptation. Therefore, we will explore other methods for topic adaptation in the following chapters.

Chapter 4

Probabilistic Adaptation with Bilingual Topic Models

In the last chapter we presented an approach to domain and topic adaptation using sparse word pair and phrase pair features and discriminative training. While we were able to show that topic triggers can be used to learn different translations of a source word or phrase under a latent topic, the end-to-end results evaluated with BLEU were not conclusive. Therefore, in this chapter we turn to generative models for topic adaptation to account for contextual information in a more principled way. We present a bilingual variant of Latent Dirichlet Allocation that learns topics over pairs of source and target phrases while learning topic-dependent, probabilistic translation probabilities at the same time. We show that this model is able to learn structure beyond the corpus level on a topically diverse French-English data set. The aim of the model is to enable *dynamic topic adaptation* for test documents of unknown origin which is a typical scenario when translating text from the web. While the model learns topic-adapted translation probabilities by marginalising over topics at the phrase level, the underlying document-level topic mixtures can also be used to compute additional topic-adapted features. We show that among the features we tested, the probabilistic translation feature yields the best performance. However, combining all topic-adapted features yields additive gains, which shows that different kinds of topical information can contribute to predicting the correct translation in a given context.

In previous literature, domains have often been loosely defined in terms of text corpora and we adopt the same definition here but distinguish *domains* from *topics*. For example, text from a news website can be defined as belonging to the *news domain*. In domain adaptation settings it is normally assumed that the data within a domain

is homogeneous in terms of style and vocabulary, though that is not always true in practice. The term *topic* on the other hand can either refer to the thematic content of a document (whether it is about politics, economy, medicine) or it can refer to a latent cluster in a topic model. Topic modelling for machine translation aims to find a match between thematic context and topic clusters. We view topic adaptation as fine-grained domain adaptation with the implicit assumption that there can be multiple distributions over translations within the same data set. If these distributions overlap, then we expect topic adaptation to help separate them and yield better translations than an unadapted system.

In this chapter, we take a new approach to topic adaptation by estimating probabilistic phrase translation features in a Bayesian fashion. The motivation is that automatically identifying topics in the training data can help to select the appropriate translation of a source phrase in the context of a document. By adapting a system to automatically induced topics, we do not have to trust data from a given domain to be uniform. We also overcome the problem of defining the level of granularity for domain adaptation. With more and more training data automatically extracted from the web with little knowledge about its content, we believe this is an important area to focus on. Translation of web sites is already a popular application for MT systems and could be helped by dynamic model adaptation. We focus on translation model adaptation to learn how words and phrases translate in a given document context without knowing from which domain the document was taken.

4.1 Related work

In this section we review some of the previous work on topic adaptation which is particularly related to the model described in this chapter. Most of the previous work using topic information for statistical machine translation used monolingual topic models. For example, Gong et al. (2010) group test sentences by topics and filter the phrase table according to a comparison of the maximum topic of each phrase pair and the test document. Gong and Zhou (2011) use the topical relevance of a target phrase, computed using a mapping between source and target side topics, as an additional feature in decoding. Axelrod et al. (2012) build topic-specific translation models from the TED corpus and select additional topic-relevant data from the UN corpus to improve coverage. None of this work has attempted to adapt probabilistic translation features or use bilingual information for topic modelling which is what we aim to do in this

chapter.

Su et al. (2012) perform phrase table adaptation in a setting where only monolingual in-domain data and parallel out-of-domain data are available. They do adapt translation probabilities but do so by scoring phrase pairs according to how relevant they are given a mapping between in-domain and out-of-domain topics. Thus, they work in a purely cross-domain setting while we aim for dynamic adaptation.

Eidelman et al. (2012) use topic-dependent lexical weights as features in the translation model. Our work is similar in that topic features are not tuned towards a target domain but towards usefulness of topic information. Hewavitharana et al. (2013) perform dynamic adaptation with monolingual topic models, encoding topic similarity between a conversation and training documents in an additional feature. Their work is similar to the work of Banchs and Costa-jussà (2011), both of which inspired our document similarity feature.

Also related is the work of Sennrich (2012a) who explore mixture-modelling on unsupervised clusters for domain adaptation and Chen et al. (2013b) who compute phrase pair features from vector space representations that capture domain similarity to the development set. Again, both of these are cross-domain adaptation approaches which require knowledge of the target domain.

Instances of multilingual topic models outside the field of MT include Boyd-Graber and Blei (2009) and Boyd-Graber and Resnik (2010) who learn cross-lingual topic correspondences but do not learn conditional distributions like our model does. In terms of model structure, our model is similar to BiTAM (Zhao and Xing, 2006) which is an LDA-style model to learn topic-based word alignments. The work of Carpuat and Wu (2007b) is similar to ours in spirit, though they predict the most probable translation in a context at the token level while our adaptation operates at the type level of a document.

4.2 Bilingual topic model over phrase pairs

Our model is based on LDA and infers topics as distributions over phrase pairs instead of over words. It is MT-specific in that the conditional dependencies between source and target phrases are modelled explicitly, and therefore we refer to it as phrasal LDA. Topic distributions learned on a training corpus are carried over to tuning and test sets by running a modified inference algorithm on the source sides of those sets. Translation probabilities are adapted separately to each source text under translation which makes

this a *dynamic* topic adaptation approach. In the following we explain our approach to topic modelling with the objective of estimating better phrase translation probabilities for data sets that exhibit a heterogeneous structure in terms of vocabulary and style. The advantage from a modelling point of view is that unlike with mixture models, we avoid sparsity problems that would arise if we treated documents or sets of documents as domains and tried to adapt models to each of these domains.

4.2.1 Latent Dirichlet Allocation

LDA is a generative model that learns latent topics in a document collection. In the original formulation, topics are multinomial distributions over words of the vocabulary and each document is assigned a multinomial distribution over topics (Blei et al., 2003). Our goal is to learn topic-dependent phrase translation probabilities and hence we modify this formulation by replacing words with phrase pairs. This is straightforward when both source and target phrases are observed but requires a modified inference approach when only source phrases are observed in an unknown test set. Different from standard LDA and previous uses of LDA for MT, we define a bilingual topic model that learns topic distributions over phrase pairs. This allows us to model the units of interest in a more principled way, without the need to map per-word or per-sentence topics to phrase pairs. Figure 4.1 shows a graphical representation of the following generative process.

For each of N documents in the collection

1. Choose topic distribution $\theta_d \sim \text{Dirichlet}(\alpha)$.
2. Choose the number of phrases pairs P_d in the document, $P_d \sim \textit{Poisson}(\zeta)$.
3. For every position d_i in the document corresponding to a phrase pair $p_{d,i}$ of source and target phrase s_i and t_i[1]:
   (a) Choose a topic $z_{d,i} \sim \text{Multinomial}(\theta_d)$.
   (b) Conditioned on topic $z_{d,i}$, choose a source phrase $s_{d,i} \sim \text{Multinomial}(\psi_{z_{d,i}})$.
   (c) Conditioned on $z_{d,i}$ and $s_{d,i}$, choose a target phrase $t_{d,i} \sim \text{Multinomial}(\phi_{s_{d,i},z_{d,i}})$.

α, β and γ are parameters of the Dirichlet distributions, which are asymmetric for $k = 0$. Our inference algorithm is an implementation of collapsed variational Bayes (CVB), with a first-order Gaussian approximation (Teh et al., 2006). It is written in

[1]Parallel documents are modelled as bags of phrase pairs.

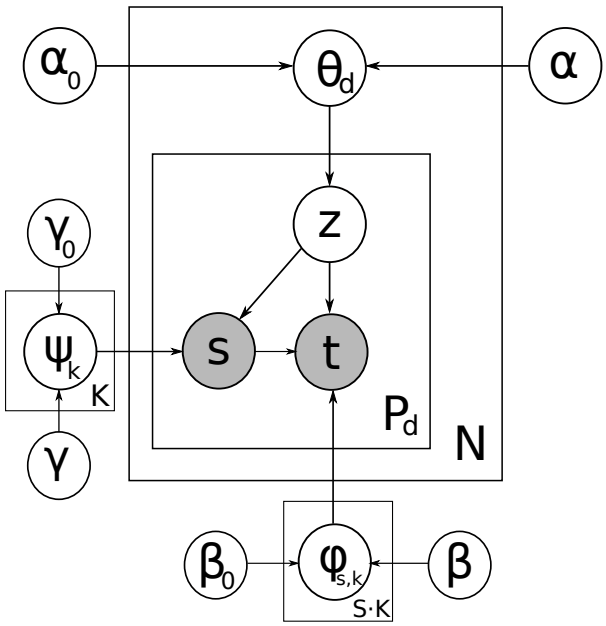


Figure 4.1: Phrasal LDA model for inference on training data: both source and target phrases are observed.

Python and uses OpenMPI for parallelization. CVB has been shown to be more accurate than standard VB and to converge faster than collapsed Gibbs sampling (Teh et al., 2006; Wang and Blunsom, 2013), with little loss in accuracy. Because we have to do inference over a large number of phrase pairs, Gibbs sampling becomes unscalable due to slow convergence as well as the frequent random number generation involved in sampling. Hence, we choose CVB which avoids this problem and is therefore more practical for our task. We parallelize inference at the document level and mix relevant counts across processes several times per training iteration.

4.2.2 Overview of training strategy

Ultimately, we want to learn translation probabilities for all possible phrase pairs that apply to a given test document during decoding. Therefore, topic modelling operates on phrase pairs as they will be seen during decoding. Given word-aligned parallel corpora from several domains, we extract lists of per-document phrase pairs produced by the extraction algorithm in the Moses toolkit (Koehn et al., 2007) which contain all phrase pairs that are consistent with the word alignment. We run CVB on the set of all training documents to learn latent topics without providing information about the domains. Using the trained model, CVB with modified inference is run on all test documents with the set of possible phrase translations that a decoder would load from

a phrase table before decoding. When test inference has finished, we compute adapted translation probabilities at the document-level by marginalising over topics for each phrase pair.

4.3 Bilingual topic inference

In the following, we provide a detailed description of the inference algorithm for both the training and testing phase. We provide an example that illustrates the difference between posterior topic distributions at the phrase level versus at the document level and show how adapted translation probabilities are computed.

4.3.1 Inference on training documents

The aim of inference on the training data is to find latent topics in the distributions over phrase pairs in each document and infer document-topic and topic-phrase pair distributions. This is done by repeatedly visiting all phrase pair positions in all documents, computing conditional topic probabilities and updating counts.

Motivation for asymmetric prior To bias the model to cluster stop word phrases in one topic, we place an asymmetric prior over the hyperparameters[2] as described in Wallach et al. (2009) to make one of the topics *a priori* more probable in every document while all the other topics remain equally probable. The reason we do not remove stop words is that because the model is defined over phrases, there is no consistent way of removing stop words without affecting phrase segmentation during decoding. One problem is that if a phrase pair is excluded from topic modelling, we need to set the adapted features to a neutral value for that phrase pair. Depending on the kind of feature, it is difficult to find such a neutral value. The main issue is that because decoding in a machine translation system allows all possible segmentations of the source sentence, phrases covering different spans of the input compete with each other. Setting the adapted features for some of the phrase pairs to zero or one, for example, would result in a systematic difference in the feature space that would be likely to have an effect on phrase segmentation. Even if we consistently excluded phrase pairs in a way that for a given source phrase all phrase pairs including this source phrase were excluded, this would still make a span covered by the source phrase more or less likely

[2]Omitted from the following equations for simplicity.

than other segmentations covering the same span, depending on how the default values for the adapted features are set. Therefore, to ensure consistency of the model we take the extra computational effort of adapting the features of all possible phrase pairs, even if a lot of them are not expected to be topically relevant.

Training procedure At the beginning of training inference, the topic mixtures for all documents are initialized by drawing for each document a distribution over topics from a Dirichlet distribution with parameter vector $\{\alpha_0, \underbrace{\alpha, \alpha, \ldots, \alpha}_{K-1 \textit{ times}}\}$. Then, we visit each position in the document collection by setting up a list of *(document, position)* tuples, shuffling it randomly and iterating over the list in sequence. At each position, we remove the set of co-occurrence counts previously added at this position, compute the new variational posterior and update the counts according to the new posterior. The counts constitute the sufficient statistics for computing the posterior distribution over the latent variables (topics) for each token (phrase pair) in each document and are defined below. As in standard LDA, the set of all possible assignments of topics to tokens in the entire collection of training documents defines a space that is too large to enumerate exhaustively and is therefore approximated by sampling.

Definition of posterior distributions For collapsed Gibbs sampling, the conditional probability of the latent variable z_i in document d at position i being assigned to topic k, given the current state of all variables (topic assignments) except $z_{d,i}$ is given by

$$\begin{aligned} P(z_i = k|\mathbf{z}^{-(d,i)}, \mathbf{s}, \mathbf{t}, d, \alpha, \beta, \gamma) \propto \\ & P(t_i|s_i, z_i = k, \mathbf{z}^{-(d,i)}, \mathbf{s}^{-(d,i)}, \mathbf{t}^{-(d,i)}, \beta) \cdot \\ & P(s_i|z_i = k, \mathbf{z}^{-(d,i)}, \mathbf{s}^{-(d,i)}, \gamma) \cdot \\ & P(z_i = k|\mathbf{z}^{-(d,i)}, d, \alpha) \end{aligned} \tag{4.1}$$

where $\mathbf{s}$ and $\mathbf{t}$ are all source and target phrases in the collection. The posterior distibution over topics is factored as the probability of a topic k given a document d, $P(z_i = k|\ldots, d, \ldots)$, the probability of a source phrase s_i given a topic k, $P(s_i|z_i = k, \ldots)$ and the probability of a target phrase t_i given a source phrase s_i and a topic k, $P(t_i|s_i, z_i = k, \ldots)$. Each of these factors can be computed as shown in equations 4.2, 4.3 and 4.4 (see Section 2.2.5 for an explanation of why this is valid)

$$P(t_i|s_i, z_i = k, \mathbf{z}^{-(d,i)}, \mathbf{s}^{-(d,i)}, \mathbf{t}^{-(d,i)}, \beta) = \frac{(c_{.,k,s,t}^{-(d,i)} + \beta)}{(c_{.,k,s,.}^{-(d,i)} + T_s \cdot \beta)} \tag{4.2}$$

$$P(s_i|z_i = k, \mathbf{z}^{-(d,i)}, \mathbf{s}^{-(d,i)}, \gamma) = \frac{(c^{-(d,i)}_{.,k,s,.} + \gamma)}{(c^{-(d,i)}_{.,k,.,.} + S \cdot \gamma)} \tag{4.3}$$

$$P(z_i = k|\mathbf{z}^{-(d,i)}, d, \alpha) \propto (c^{-(d,i)}_{d,k,.,.} + \alpha) \tag{4.4}$$

where $c^{-(d,i)}_{.,k,s,t}$, $c^{-(d,i)}_{.,k,s,.}$ and $c^{-(d,i)}_{d,k,.}$ are co-occurrence counts of topics with phrase pairs, source phrases and documents respectively, $c^{-(d,i)}_{.,k,.}$ is a topic occurrence count, T_s is the number of possible target phrases for a given source phrase and S is the total number of source phrases. The posterior distribution in Equation 4.1 can then be written as

$$P(z_i = k|\mathbf{z}^{-(d,i)}, \mathbf{s}, \mathbf{t}, d, \alpha, \beta, \gamma) \propto \frac{(c^{-(d,i)}_{.,k,s,t} + \beta)}{(c^{-(d,i)}_{.,k,s,.} + T_s \cdot \beta)} \cdot \frac{(c^{-(d,i)}_{.,k,s,.} + \gamma)}{(c^{-(d,i)}_{.,k,.,.} + S \cdot \gamma)} \cdot (c^{-(d,i)}_{d,k,.,.} + \alpha). \tag{4.5}$$

By modelling the dependencies of source and target phrases on topics separately as $P(t_i|s_i, z_i = k, ..)$ and $P(s_i|z_i = k, ..)$, we collect the necessary statistics to compute conditional translation probabilities that are also conditioned on topics. The factorisation also enables us to put different priors on these distributions. For example, we want a sparse distribution over target phrases for a given source phrase and topic to express our translation preference under each topic. But the distribution over source phrases for a given topic should be relatively less sparse because there are many more possible source phrases that can occur under a specific topic. Of course, the topic does have an influence on the selection of the source phrase as well.

Another way of decomposing the posterior distribution in Equation 4.1 is shown in Equation 4.6 where P is the total number of phrase pairs in the training data. In this model, which is more similar to standard LDA in terms of its formulation, pairs of source and target phrases are generated jointly given topic k, as expressed by the probability $P(t_i, s_i|z_i = k, \dots)$.

$$\begin{aligned} &P(z_i = k|\mathbf{z}^{-(d,i)}, \mathbf{s}, \mathbf{t}, d, \alpha, \beta, \gamma) \propto \\ &\quad P(t_i, s_i|z_i = k, \mathbf{z}^{-(d,i)}, \mathbf{s}^{-(d,i)}, \mathbf{t}^{-(d,i)}, \beta) \cdot \\ &\quad P(z_i = k|\mathbf{z}^{-(d,i)}, d, \alpha) \propto \\ &\quad \frac{(c^{-(d,i)}_{.,k,s,t} + \beta)}{(c^{-(d,i)}_{.,k,.,.} + P \cdot \beta)} (c^{-(d,i)}_{d,k,.,.} + \alpha) \end{aligned} \tag{4.6}$$

While generating phrase pairs given topics makes sense intuitively, we lose the conditional relationship between source and target phrases in the joint formulation. Since ultimately we are interested in conditional translation probabilities to perform topic adaptation, the first factorisation is more appropriate for our purpose.

For CVB with a first-order Gaussian approximation, the counts in Equation 4.5 are replaced by their means. This means that instead of adding a count of 1 for each occurrence of a topic in a document as in sampling, we add the topic proportion under the variational posterior, which is the expected count of the topic under the variational posterior.

The conditional probability of topic z_i given the current state of all variables except $z_{d,i}$ is then given by

$$P(z_{d,i} = k|\mathbf{z}^{-(d,i)}, \mathbf{s}, \mathbf{t}, d, \alpha, \beta, \gamma) \propto \frac{(\mathbb{E}_{\hat{q}}[c_{.,k,s,t}^{-(d,i)}] + \beta)}{(\mathbb{E}_{\hat{q}}[c_{.,k,s,.}^{-(d,i)}] + T_s \cdot \beta)} \cdot \frac{(\mathbb{E}_{\hat{q}}[c_{.,k,s,.}^{-(d,i)}] + \gamma)}{(\mathbb{E}_{\hat{q}}[c_{.,k,.}^{-(d,i)}] + S \cdot \gamma)} \cdot (\mathbb{E}_{\hat{q}}[c_{d,k,.}^{-(d,i)}] + \alpha) \tag{4.7}$$

where $\mathbb{E}_{\hat{q}}$ is the expectation under the variational posterior which is computed as shown in Equation 2.34. The similarity of Equations 4.5 and 4.7 shows that "CVB is indeed the mean field version of collapsed Gibbs sampling" (Teh et al., 2006).

Runtime The algorithm stops when the variational posterior has converged for all documents or after a maximum of 100 iterations. Empirical results showed that the performance after 50 iterations of inference is often very similar to the performance after 100 iterations. To speed up training and to save memory, singleton phrase pairs that do not occur in the development or test sets are removed from the document collection[3].

Training a model with 20 topics on 10 cores of 2.67 GHz for 50 iterations took ∼16 hours (∼38 hours for a model with 50 topics). Note that this includes writing out model and topic mixture files after every block of 10 iterations which is very time consuming. Writing out files during training could be avoided since it is only used for debugging purposes and in case the training process is interrupted. Note also that all runtimes mentioned in this thesis have been measured under conditions where resources are shared with other jobs and can therefore fluctuate highly.

[3]Obviously, this cannot be done under real test conditions where the test set is not known in advance. However, we could perform a similar approximation by falling back to the baseline translation probabilities for all singleton phrase pairs in the training data.

CVB versus Gibbs sampling In Gibbs sampling, the topic assignment at every position in each document has to be stored and is updated once a new assignment has been sampled from the posterior topic distribution. For CVB, instead of a topic assignment we need to store the entire posterior topic distribution for each position. At each step in the iteration, this stored distribution is used to remove fractional topic counts before computing the new posterior topic distribution. After computing the new posterior, the counts are updated using the new fractional topic counts, which are the topic probabilities in the normalised posterior topic distribution. Teh et al. (2006) describe that the memory requirements of CVB are reduced considerably by keeping only one copy of the variational posterior for each pair of document and phrase pair type, instead of keeping a copy for every phrase pair token. While all results reported in this chapter are derived from a model that keeps track of one posterior distribution per token, an additional experiment showed that keeping only one posterior distribution per type achieved almost the same performance on the downstream translation task, with a decrease in BLEU of ~ 0.1.

4.3.2 Hyperparameter optimisation

To avoid a grid search over the hyperparameters of the model, we experimented with a fixed-point update (Minka, 2012; Heinrich, 2009) to optimise the parameters of the Dirichlet priors (α, β, γ) after every complete training iteration. For unconstrained α (analogously for β, γ) the update is defined as

$$\alpha'_k = \alpha_k \cdot \frac{\sum_d [\psi(C_{d,k} + \alpha) - \psi(\alpha)]}{\sum_d [\psi(C_d + \sum_k \alpha_k) - \psi(\sum_k \alpha_k)]} \tag{4.8}$$

where the right-hand side term is the maximum of the gradient of the log-likelihood P(D|α) (Equations 54 and 55 of Minka (2012)). ψ is the digamma function which is defined as

$$\psi(x) = \frac{d \log \Gamma(x)}{dx} \tag{4.9}$$
$$\Gamma(x) = (x-1)!$$

The fixed-point update iteratively maximises the log-likelihood of the data. Similarly, Heinrich defines an update for symmetric Dirichlet distributions as

$$\alpha' = \alpha \cdot \frac{\sum_d \sum_k [\psi(C_{d,k} + \alpha) - \psi(\alpha)]}{K \cdot \sum_d [\psi(C_d + \sum_k \alpha_k) - \psi(\sum_k \alpha_k)]} \tag{4.10}$$

which corresponds to Equation 84 of Heinrich (2009). Because in our model all hyperparameters are symmetric except for k=0, we use Equation 4.8 to update $\alpha_0, \beta_0, \gamma_0$ and a modification of Equation 4.10 to update all other parameters as shown below

$$\alpha' = \alpha \cdot \frac{\sum_d \sum_{k,k\neq 0}[\psi(C_{d,k}+\alpha) - \psi(\alpha)]}{(K-1)\cdot \sum_d[\psi(C_d + \sum_k \alpha_k) - \psi(\sum_k \alpha_k)]}. \tag{4.11}$$

We initialised the values to α_0 = 2.0, $\beta_0 = \gamma_0$ = 1e-08, α = 0.5, β = 0.1, γ = 0.0001 and optimised them as described above. The motivation behind the parameter settings for the asymmetric part of the priors is that we want topic 0 to be a priori more probable in each document than all other topics. Therefore, we assign it a higher pseudo count. But we set the pseudo counts for β_0, γ_0 to a very small value to avoid that the model learns to assign high probability to all source and target phrases under topic 0. Setting $\alpha = 0.5$ induces sparsity while allowing the model to assign mass to more than one topic for a given document. We want both β and γ to be sparse as well and want the pseudo counts to reflect the number of possible events under each distribution. Assuming that a source phrase has around 10 reasonable translations on average, we set β to $\frac{1}{10}$. There are about 680K source phrases in our training data, so under a uniform distribution the probability would be about 1e-06, but since we want less sparsity than for β, we set γ to 1e-04. Table 4.1 shows the hyperparameter values after 50 iterations of training inference for the setups described in Section 4.6 and Section 4.7. While the relationship between α and β in the initialisation is preserved in the learned values, the model consistently chooses smaller values for β than for γ which indicates that sparsity in the distributions over target phrases is more important than sparsity in the distributions over source phrases for a given topic.

Setup	α_0	β_0	γ_0	α	β	γ
3 domains	5.137	2.34e-09	0.092	0.155	0.018	0.174
Commoncrawl	8.879	4.83e-10	0.067	0.144	0.036	0.097

Table 4.1: Hyperparameters of pLDA model after 50 iterations of training inference (k=100). The two setups refer to the experiments in Section 4.5.1 and Section 4.7.1.

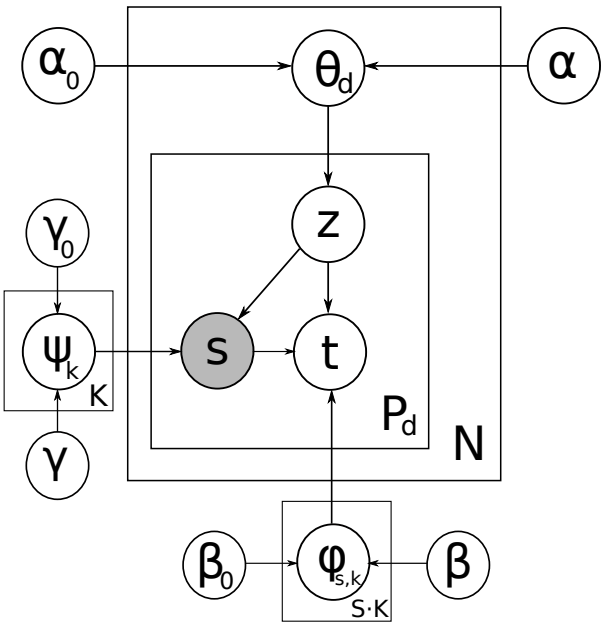


Figure 4.2: Phrasal LDA model for inference on development and test data: source phrases are observed, target phrases are unobserved.

4.3.3 Inference on tuning and test documents

In order to compute phrase translation probabilities for tuning and test documents, we have to deal with missing target phrases[4] and therefore, the model is adapted as shown in Figure 4.2 where target phrases are no longer observed. To account for the missing target phrases, the variational posterior for test inference changes as shown in Equation 4.12 which computes the joint posterior distribution over topics k and target phrases $t_{i,j}$, given the source phrase s_i and the test document d.

$$P(z_{d,i} = k, t_{i,j} | \mathbf{z}^{-(d,i)}, \mathbf{s}, \mathbf{t}^{-(d,i)}, d, \alpha, \beta, \gamma) \propto$$
$$\frac{(\mathbb{E}_{\hat{q}}[n^{-(d,i)}_{.,k,s,t_j}] + \beta)}{(\mathbb{E}_{\hat{q}}[n^{-(d,i)}_{.,k,s,.}] + T_s \cdot \beta)} \cdot \frac{(\mathbb{E}_{\hat{q}}[n^{-(d,i)}_{.,k,s,.}] + \gamma)}{(\mathbb{E}_{\hat{q}}[n^{-(d,i)}_{.,k,.}] + S \cdot \gamma)} \cdot (\mathbb{E}_{\hat{q}}[n^{-(d,i)}_{d,k,.}] + \alpha) \tag{4.12}$$

Because this distribution ranges over all possible target phrases $t_{i,j}$ for a given source phrase, the size of the support changes from K to $K \cdot T_s$. While during training inference we compute a distribution over topics for each source-target pair, in test inference we can use the posterior to marginalise out the topics and get a distribution over target phrases for each source phrase. In fact, this gives us the adapted transla-

[4]Strictly speaking, we only have missing targets for test documents, but to have the same setup for tuning and testing, we treat the targets of development sets as unobserved, too.

tion probabilities for a given document. Conversely, we can marginalise out the target phrases and get a distribution over topics for each source phrase. Inspecting this topic distribution for a given source phrase tells us whether the model considers the translation of this phrase topically relevant or not (as shown by the amount of mass assigned to topic 0) and if topically relevant, which topics are likely given the source phrase. This topic distribution effectively decides which translations of a source phrase are likely in a given document: those that are likely given the source phrase and likely under the topics favoured by the source phrase.

Search space We use the Moses decoder to produce lists of translation options for each document in the tuning and test sets. These lists comprise all phrase pairs that will enter the search space at decoding time. By default, only 20 target phrases per source phrase are loaded from the phrase table[5], so in order to allow for new phrase pairs to enter the search space and for translation probabilities to be computed more accurately, we allow for up to 200 target phrases per source. For each source sentence, we consider all possible phrase segmentations and applicable target phrases. Unlike in training, we do not iterate over all phrase pairs in the list but over blocks of up to 200 target phrases for a given source phrase.

Initialisation Before running test inference, all count variables are initialized to the values at the end of training inference. Test documents are initialized by iterating over all source positions and updating counts according to the variational posterior used for inference.

Runtime The algorithm stops when all marginal, forward phrase translation probabilities have converged though in practice we stopped earlier to avoid overfitting. Empirical results showed that between 5-10 iterations are sufficient to infer reliable topic mixtures (according to the resulting translation quality).

The runtime of the algorithm for test inference is mostly dominated by the time taken for topic inference, but also depends on the number and type of the adapted features. The runtime for 10 iterations of topic inference and computing only the probabilistic feature $P(e|f,d)$ for a model with 20 topics was $\sim$3.5 hours for all test documents, processing documents in parallel using 10 cores of 2.67 GHz. When computing all adapted features, the runtime increased to a total of $\sim$ 4.75 hours.

[5]This pruning step depends on the feature weights provided in the decoder configuration file.

The processing time grows linearly with the size of the filtered phrase table for each test document and the number of topics, thus the complexity is $O(|S| \cdot |T_s| \cdot K)$ where S is the number of source phrases, T_s is the number of target phrases per source phrase and K is the number of topics. Including the additional adapted features changes the complexity to $O(|S| \cdot |T_s| \cdot K + |V_t| \cdot K)$ where V_t is the target language vocabulary used in the phrase table.

Fixing training distributions For test inference, instead of updating all count statistics used to compute the posterior topic distributions, we keep all co-occurrence counts of topics, source and target phrases fixed. That is, the only count being updated during test inference is the co-occurrence count of documents and topics, $c_{d,k,\cdot}^{-(d,i)}$, in Equation 4.12. The motivation is that since we do not observe target phrases during test inference, we cannot learn anything new about how phrases translate in a given context and neither about the general distribution over topics. Therefore, updating only $c_{d,k,\cdot}^{-(d,i)}$ amounts to merely carrying over topical information from training to each specific test document, depending on the source phrases that occur in that test document and the possible target phrases it can be translated into. Empirical results showed that keeping the topic model fixed in this way during test inference slightly improves results on the downstream translation task.

4.3.4 Phrase translation probabilities

After topic inference on the tuning and test data, the forward translation probabilities $P(t|s,d)$ are computed. This is done separately for every document d because we are interested in the translation probabilities that depend on the inferred topic proportions for a given document. For every document, we iterate over source positions $p_{d,i}$ and use the current variational posterior to compute $P(t_{i,j}|s_i,d)$ for all possible target phrases by marginalising over topics:

$$P(t_{i,j}|s_i,d) = \sum_k P(z_i = k, t_{i,j}|\mathbf{z}^{-(d,i)}, \mathbf{s}, \mathbf{t}^{-(d,i)}, d) \tag{4.13}$$

This is straightforward because during test inference the variational posterior is normalised to a distribution over topics and target phrases for a given source phrase. If a source phrase occurs multiple times in the same document, the probabilities are averaged over all occurrences:

$$P(t_{i,j}|s_i,d) = \frac{\sum_{s=s_i} P(t_{i,j}|s_i,d)}{|\{s|s=s_i\}|} \tag{4.14}$$

4.3.5 Inverse translation features

The inverse translation probabilities $P(s_{i,j}|t_i,d)$ can be computed using the expected counts of all quantities in the variational posterior at the end of an iteration. We use this approximation because we do not have variational posteriors for all pairs of source and target phrases in a test document. The decoding search space contains only the source phrases that occur on the source side of a test document, but in order to compute inverse translation probabilities we need to consider all source phrases $s_{i,j}$ that could have produced a given target phrase t_i. Similar to before, we can marginalise over topics k to get the inverse translation probabilities:

$$P(s_{i,j}|t_i,d) = \sum_k P(z_i = k, s_{i,j}|\mathbf{z}^{-(d,i)}, \mathbf{s}^{-(d,i)}, \mathbf{t}, d) \tag{4.15}$$

We performed experiments with both the adapted forward and inverse translation feature, $P(t|s,d)$ and $P(s|t,d)$ and found no improvement when including the inverse adapted feature. While we have not investigated this in detail, it could be related to the conditional formulation of the bilingual topic model. The model learns the topical structure needed to differentiate translations from the source to the target language. Translation ambiguities are not symmetric, though, and in the inverse translation direction, a different clustering of phrase pairs may be necessary to capture ambiguities.

4.3.6 Posterior topic mixtures at the phrase level

An important detail of the phrasal topic model is that even though topic mixtures are assigned at the document level, the posterior distributions over topics are computed at the phrase pair level (in training) or source phrase level (in testing), and the adapted translation feature $P(e|f,d)$ is computed from this posterior distribution, not from the document topic mixtures. This is important because the document topic mixture is just one factor in the posterior distribution and the topic distributions at the phrase level are much more precise than at the document level, and can also vary quite drastically.

Figure 4.3 shows an example of the difference between document topic distributions $P(z|d)$ and phrase pair topic distributions $P(z|s,t,d)$ for two translations of the ambiguous French source word *mars*. The translations *mars* $\rightarrow$ *mars* and *mars* $\rightarrow$

P(z|d=666) =
0.3520 0.0004 0.0002 0.0003
0.0003 0.0018 0.0041 0.0002
0.0013 0.0002 0.0008 **0.1478**
0.0002 0.0004 0.0003 0.0003
0.4183 0.0046 0.0002 0.0663

P(z|d=3262) =
0.2571 0.0665 0.0103 0.0017
0.0012 **0.5450** 0.0307 0.0016
0.0040 0.0015 0.0077 0.0012
0.0013 0.0269 0.0035 0.0065
0.0014 0.0287 0.0015 0.0017

P(z|src=*mars*, trg=*mars*,d=666) =
6.1e-03 1.0e-06 3.1e-07 1.0e-11
5.9e-12 3.8e-06 5.3e-10 4.1e-07
5.3e-11 1.8e-11 7.8e-12 5.3e-06
1.9e-12 1.1e-11 4.8e-08 1.2e-11
9.9e-01 6.9e-06 3.4e-12 1.2e-04

P(z|src=*mars*, trg=*march*,d=3262) =
3.8e-02 2.9e-02 2.7e-05 4.8e-11
1.6e-11 **9.2e-01** 3.0e-09 2.2e-06
1.3e-10 9.0e-11 5.9e-11 2.8e-08
1.1e-11 6.3e-10 6.4e-07 2.1e-10
2.6e-06 9.1e-03 2.2e-11 3.6e-06

Figure 4.3: Document-topic distributions $P(z|d)$ for documents 666 and 3262 compared to phrase-topic distributions $P(z|src,trg,d)$ for two different translations of the French source word *mars*. Left: distributions for a document containing the phrase pair *mars* $\rightarrow$ *mars*. Right: distributions for a document containing the phrase pair *mars* $\rightarrow$ *march*.

march correspond to two different senses of the source word and would likely occur in different document context.

Figure 4.3 shows that in both document topic distributions (each associated with a specific document d) some of the mass is assigned to topic 0 (0.352023 and 0.257070) which groups common phrase pairs. Both documents also have peaks at other topics, capturing the actual content of the document. In the first example, there are peaks at topic 11 and topic 16 (marked in bold) and in the second example there is a peak at topic 5. The phrase pair topic distributions correspond to the topic mixtures for the phrase pairs *mars* $\rightarrow$ *mars* and *mars* $\rightarrow$ *march* occurring in the respective documents. In these distributions, the mass is shifted away from topic 0 and almost all of the mass is accumulated at the content topics (topic 16 in the first example and topic 5 in the second example). This shows that using phrase pair topic distributions for adaptation is more precise than using document topic distributions, and that having a dedicated topic to account for common phrase pairs helps to keep the actual topic distributions cleaner.

4.4 More topic-adapted features

Inspired by previous work on topic adaptation for SMT, we add three additional topic-adapted features to our model. All of these features make use of the topic mixtures learned by our bilingual topic model. The first feature is an adapted lexical weight, similar to the features in the work of Eidelman et al. (2012). Our feature is different in that we marginalize over topics to produce a single adapted feature where $v[k]$ is the k^{th} element of a document topic vector for document d and $w(t_j|s_i,k)$ is a topic-dependent word translation probability that depends on the word alignment $\mathbf{a}$:

$$lex(t|s,d) = \prod_{j=1}^{|t|} \frac{1}{|\{i|(i,j) \in \mathbf{a}\}|} \sum_{\{i|(i,j)\in\mathbf{a}\}} \underbrace{\sum_k w(t_j|s_i,k) \cdot v[k]}_{w(t_j|s_i)} \tag{4.16}$$

Eidelman et al. (2012) instead include K new features (with K the number of topics) for the most probable topic, the second most probable topic and so on. Thus, they learn K additional feature weights that capture the usefulness of the most probable topic for a given example down to the least probable topic, independent of which specific topic provides the feature values for a given example. Because we sum out the latent topics, we learn only one additional feature weight which captures the usefulness of the entire topic distribution. This simplifies learning feature weights, especially for large numbers of topics.

The second feature is a target unigram feature similar to the lazy MDI adaptation of Ruiz and Federico (2012), which is a technique to adapt a language model without changing the background language model which would require computing normalisation terms for every adaptation context. Instead, lazy MDI adaptation adds unigram ratio scaling terms only for unigrams w_i that appear in the translation options for a given test instance, and adds these scaling terms as features in the log-linear translation model, as shown below

$$P_{LM\_adapt}(t|s) = \prod_{i=1}^{|t|} \hat{\alpha}(w_i)^{\gamma} \tag{4.17}$$

$$\hat{\alpha}(w_i) = f\left(\frac{P_A(w_i)}{P_B(w_i)}\right) \tag{4.18}$$

$$f(x) = \frac{2}{1+\frac{1}{x}}, \quad x > 0 \tag{4.19}$$

where $|t|$ is the length of a translation option and γ is a weight in the log-linear model. P_A is the probability according to an adaptation text and P_B is the probability under a background language model. The function f(x) in Equation 4.19 is a fast sigmoid approximation that bounds x to the interval $[0,2]$ (compare Equation 14 in Ruiz and Federico (2012)).

We apply lazy MDI adaptation by adding a feature that multiplies unigram ratio scaling terms for all words in a target phrase, including an additional term that measures the topical relevance of a target word w_i (Equation 4.20). $P_{doc}(w_i)$ is the adapted unigram probability for a given document, summed over the document topic mixture and $P_{topic0}(w_i)$ is the unigram probability under topic 0. While the first term in Equation 4.20 captures the difference between the adapted unigram probability and the baseline probability, the relevance term captures the intuition that a word is not considered topically relevant if it has a high probability of occurring under topic 0. Therefore, larger ratios of P_{doc} to P_{topic0} indicate higher topical relevance.

$$trgUnigrams_t = \prod_{i=1}^{|t|} \underbrace{f(\frac{P_{doc}(w_i)}{P_{baseline}(w_i)})}_{lazy\ MDI} \cdot \underbrace{f(\frac{P_{doc}(w_i)}{P_{topic0}(w_i)})}_{relevance} \tag{4.20}$$

$$P_{doc}(w_i) = \frac{\sum_k P(w_i|k) \cdot v[k]}{\sum_{w_{i'}} \sum_k P(w_{i'}|k) \cdot v[k]} \tag{4.21}$$

$$P_{topic0}(w_i) = \frac{P(w_i|k=0)}{\sum_{w_{i'}} P(w_{i'}|k=0)} \tag{4.22}$$

The third feature is a document similarity feature, similar to the semantic feature described by Banchs and Costa-jussà (2011):

$$docSim_t = \max_i (1 - JSD(v_{train\ doc_i}, v_{test\ doc})) \tag{4.23}$$

where $v_{train\_doc_i}$ and v_{test_doc} are document topic vectors of training and test documents. Because topic 0 captures phrase pairs that are common to many documents, we exclude it from the topic vectors before computing similarities. While our feature compares document topic vectors of phrase pairs, Banchs and Costa-jussà (2011) compare vectors capturing the source sentence context of a phrase pair using Latent Semantic

Indexing. The final similarity score is the maximum of the similarity scores computed for the test context and every occurrence of the phrase pair in the training data. We use the symmetrised Jensen-Shannon divergence (JSD) to compare topic mixtures as proposed by Steyvers and Griffiths (2007) (for $\log_2$, JSD $\in [0,1]$) which is based on the Kullback-Leibler (KL) divergence and defined as

$$\begin{aligned} JSD(p,q) &= \frac{1}{2}\Big(KL\big(p,(p+q)/2\big) + KL\big(q,(p+q)/2\big)\Big) \\ KL(p,q) &= \sum_{x=1}^{K} p(x)\log_2\frac{p(x)}{q(x)}. \end{aligned} \tag{4.24}$$

We experimented with another topic-adapted feature that compares the topic mixture for a given test source phrase to the average topic mixture of a phrase pair. Intuitively, this measures the topical similarity of a source phrase in a given document context to all of its possible translations, represented by all applicable phrase pairs where the source phrase matches. The phrase similarity feature is defined as

$$\begin{aligned} phrSim_t &= (1 - JSD(v_{pp_t}, v_{source\ phrase_d})) \\ v_{pp_t}[k] &= \frac{count_{s,t,k}}{\sum_k count_{s,t,k}} \end{aligned} \tag{4.25}$$

where $v_{source\ phrase_d}$ is the average topic mixture for a source phrase in document d according to the marginal topic distribution of Equation 4.12. $v_{pp_t}[k]$ is the normalised average occurrence of a phrase pair pp_t under topic k, as given by the co-occurrence counts collected during training inference. The comparison of the source phrase topic vector to the phrase pair topic vector measures how well a given phrase pair matches the topic distribution of the source phrase in the given document context. However, this similarity feature did not improve the performance when combined with the other four adapted features and received a very low or negative weight when replacing the document similarity feature in the log-linear model. We are not sure why this feature performs poorly but we revisit the idea behind this feature in Chapter 5. The similarity feature described there is similar in that the comparison involves a phrase topic mixture, but differs in the way the phrase pair vector is computed. Further, the phrase pair vector is compared to a representation of the test context instead of a source phrase vector, even though both are indicative of the topical preference of the test context.

4.4.1 Feature combination

We tried integrating the four topic-adapted features separately and in all possible combinations. As we will see in the results section, while all features improve over the baseline in isolation, the adapted translation feature $P(t|s,d)$ is the strongest feature. For the features that have a corresponding feature in the baseline model ($P(t|s,d)$ and $lex(t|s,d)$), we experimented with either adding or replacing them in the log-linear model. We found that while adding the features worked well and yielded close to zero weights for their baseline counterparts after tuning, replacing them yielded better results in combination with the target unigram and document similarity features. We believe the reason could be that a smaller total number of features in the phrase table is easier to optimize.

4.5 Experimental setup

In this section we describe the experimental setup for the evaluation of the phrasal LDA model on a machine translation task. We first describe the baseline and domain-adaptation benchmark systems in the context of a mixed data set containing three French-English corpora. The experimental results on this data set are described in Section 4.6. We carried out a further evaluation on a related data set that contains data from only one corpus (Commoncrawl) in order to compare to the domain-adaptation benchmark systems when trained on unsupervised clusters of the training data. The results of this evalution are described in Section 4.7. In the first evaluation, the domain-adaptation benchmarks have more information than the pLDA system, because they make use of their knowdledge about domain boundaries (both in training and in testing). In the second evaluation, the comparison is fairer because none of the systems has any knowledge about the internal structure of the data set.

4.5.1 Data and baselines

Our first set of experiments was carried out on a mixed data set, containing the TED corpus (Cettolo et al., 2012), parts of the News Commentary corpus (NC) and parts of the Commoncrawl corpus (CC) from the WMT13 shared task (Bojar et al., 2013) as described in Table 4.2. We were guided by two constraints in choosing our data set: 1) the data has document boundaries and the content of each document is assumed to be topically related, 2) there is some degree of topical variation within each data

Data	Mixed		CC	NC	TED
Train	354K	(6450)	110K	103K	140K
Dev	2453	(39)	818	817	818
Test	5664	(112)	1892	1878	1894

Table 4.2: Number of sentence pairs and documents (in brackets) in the French-English data sets. The training data has 2.7M English words per domain.

set. In order to compare to domain adaptation approaches, we chose a setup with data from different corpora. We believe that the broad range of this data set makes it a suitable testbed for topic adaptation. In order to abstract away from adaptation effects that concern tuning of length penalties and language models, we use a mixed tuning set containing data from all three domains and train one language model on the concatenation of (equally sized) target sides of the training data. Word alignments are trained on the concatenation of all training data and fixed for all models.

Our baseline (ALL) is a phrase-based French-English system trained on the concatenation of all parallel data. It was built with the Moses toolkit (Koehn et al., 2007) using the 14 standard core features including a 5gram language model. Translation quality is evaluated on a large test set[6], using the average feature weights of three optimisation runs with PRO (Hopkins and May, 2011). We use the mteval-v13a.pl[7] script to compute case-insensitive BLEU. As domain-aware benchmark systems, we use the phrase table fillup method (FILLUP) of Bisazza et al. (2011) which preserves the translation scores of phrases from the IN model and the linear mixture models (LIN-TM) of Sennrich (2012b) (both available in the Moses toolkit). For both systems, we build separate phrase tables for each domain and use a wrapper to decode tuning and test sets with domain-specific tables. Specifically, for a given domain, we distinguish between in-domain and out-of-domain data, where the out-of-domain data comes from the other two domains. For the FILLUP model, the scores are taken from the in-domain phrase table for all phrase pairs present in the in-domain table and from an out-of-domain phrase table for all remaining pairs. The results of these systems can be found in Section 4.6.2, Table 4.8. Both benchmark systems have an advantage over our model because they are aware of domain boundaries in the test set. LIN-TM adapts phrase table features in both translation directions while we only adapt the forward features,

[6]The test set is about the size of two standard WMT test sets.

[7]ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl

Data	Mixed	CC	NC	TED
IN	26.77	18.76	**29.56**	**32.47**
ALL	26.86	**19.61**	29.42	31.88

Table 4.3: Average BLEU of in-domain (IN) and baseline (ALL).

but add two additional features with no correspondence in the baseline models.

Table 4.3 shows BLEU scores of the baseline system as well as the performance of three in-domain models (IN) tuned under the same conditions. For the IN models, every portion of the test set is decoded with a domain-specific model. Results on the test set are broken down by domain but also reported for the entire test set (mixed). For TED and NC, the in-domain models perform better than ALL, while for CC the all-domain model improves quite significantly over IN.

4.5.2 General properties of the data sets

In this section we analyse some internal properties of our three data sets that are relevant for adaptation. All of the scores were computed on the sets of source side tokens of the test set which were limited to contain content words (nouns, verbs, adjectives and adverbs). The test set was tagged with the French TreeTagger (Schmid, 1994). The top of Table 4.4 shows the average Jensen-Shannon divergence of each in-domain model in comparison to the all-domain model, which is an indicator of how much the distributions in the IN model change when adding out-of-domain data. Likewise, Rank1-diff gives the percentage of word tokens in the test set where the preferred translation according to $p(e|f)$ changes between IN and ALL. These are the words that are most affected by adding data to the IN model. Both numbers show that for Commoncrawl, the IN and ALL models differ more than in the other two data sets. According to the JS divergence between NC-IN and ALL, translation distibutions in the NC phrase table are most similar to the ALL phrase table. Table 4.5 shows the average JSD for each IN model compared to a model trained on half of its in-domain data. This score gives an idea of how diverse a data set is, measured by comparing distributions over translations for source words in the test set[8]. According to this score, Commoncrawl is the most diverse data set and TED the most uniform. Note however, that these divergence scores do not provide information about the relative quality of the systems under comparison.

[8]Ideally, we would compute this score over several samples of the in-domain data.

Models	Avg JSD	Rank1-diff
CC-IN vs ALL	0.17	18.4%
NC-IN vs ALL	0.13	13.3%
TED-IN vs ALL	0.15	10.8%

Table 4.4: Average JSD of IN vs. ALL models. Rank1-diff: % phrase table entries where preferred translation changes.

Models	Avg JSD
CC-half vs CC-full	0.17
NC-half vs NC-full	0.09
TED-half vs TED-full	0.07

Table 4.5: Average JSD of in-domain models trained on half vs. all of the data.

For CC, the ALL model yields a much higher BLEU score than the IN model and it is likely that this is due to noisy data in the CC corpus. In this case, the high divergence is likely to mean that distributions are corrected by out-of-domain data rather than being shifted away from in-domain distributions.

4.5.3 Topic-dependent decoding

The phrase translation probabilities and additional features described in the last two sections are used as features in the log-linear translation model in addition to the baseline translation features. When combining all four adapted features, we replace $P(t|s)$ and $lex(t|s)$ by their adapted counterparts. We construct separate phrase tables for each document in the development and test sets and use a wrapper around the decoder to ensure that each input document is paired with a configuration file pointing to its document-specific translation table. The decoder runs with multiple threads on a document, but the documents are decoded in sequence so that only one phrase table has to be loaded at a time. 100best lists of translations are collected in temporary files in order for the indices to be adjusted after concatenating the 100best lists for all documents. Using the wrapped decoder we can run parameter optimisation (PRO) in the usual way to get one set of tuned weights for all test documents.

```
# sum updates to count variable
sum_updates_j_k = numpy.zeros((self.col.J_total, self.K), dtype=..)
comm.Reduce(self.updates_j_k, sum_updates_j_k, op=MPI.SUM)

# updates_j_k: broadcast sum of updates
sum_updates_j_k = comm.bcast(sum_updates_j_k, root=0)

# updates_j_k: compute new values of count variables
self.count_j_k += (sum_updates_j_k - self.updates_j_k)
```

Figure 4.4: OpenMPI instructions for summing a variable between processors (Reduce) with operator MPI.SUM and broadcasting the result back from the root processor (bcast). Local variables are updated by adding the updated counts from all other processors.

4.5.4 Implementation details of parallelisation

The topic modelling algorithm described in this chapter was implemented in python, using the *mpi4py* library to integrate OpenMPI functionality as well as the *numpy* library to support vectors and matrices. Parallelisation with OpenMPI is relatively straightforward to implement. The python program is started as an argument of the executable `mpirun`, specifying the number of processors needed with `-n`: `mpirun -n 8 plda.py <args>`.

The communicator `MPI_COMM_WORLD` then provides access to all available processors. Unless specified otherwise, all processors execute the same code in parallel. Operations that need to be performed only once, for example writing model files to disk, can be nested in an if-statement that selects a single processor by its `rank`. Phrasal LDA is parallelised at the document level, which means that each processor operates on a fraction of the document collection. Because many phrase pairs occur in multiple documents, a parallelised model is likely to be inaccurate because each processor misses co-occurrence counts from other documents with relevant phrase pairs. To ensure that the resulting model is as accurate as possible, all counts that are not document-specific are shared across all processors at specified intervals. An example is shown in Figure 4.4, where updates to the co-occurrence variable `self.count_j_k` are summed in the local variable `sum_updates_j_k`, then broadcast to all processors by the root process and finally added to the count variable of each individual processor.

```
[GENERAL]
decoder = "$working-dir/runMoses.docInput.perl"
...
[TUNING]
decoder-settings = ".. -feature-type replace -text-type dev -model-dir
adapted_ttables/ -model 3domains.k50.exclSgl.hypOpt.asym -replace-BL"
...
[EVALUATION]
decoder-settings = ".. -feature-type replace -text-type test -model-dir
adapted_ttables/ -model 3domains.k50.exclSgl.hypOpt.asym"
...
```

Figure 4.5: Example of integrating document-wise decoding with topic-adapted models in EMS configuration file.

4.5.5 Integration with Moses decoder

The training process of the pLDA models is decoupled from the training of other model components in the translation system. At test time, however, we need to produce adapted translation probabilities for each input document. This step could in principle be integrated more tightly with the Moses decoder, however this is not strictly necessary because the adapted features can be precomputed before decoding the test set. We take a simple approach where topic adaptation and feature computation are run outside of Moses to produce document-specific, adapted phrase tables.

We use a wrapper script around the Moses decoder to load input documents along with their adapted phrase tables and configuration files. The wrapper makes sure that input documents are decoded in sequence and the outputs are merged. It also takes care of applying tuned feature weights to each document-specific configuration file. In the configuration file for Moses' Experiment Management System (EMS)[9], we simply have to specify the wrapper script and a few decoding options for tuning and testing to select the appropriate models. An example is shown in Figure 4.5.

[9] http://www.statmt.org/moses/?n=FactoredTraining.EMS

4.6 Evaluation on mixed domain data

In this section we present experimental results with phrasal LDA on the mixed domain data set. We show BLEU scores in comparison to a baseline system and two domain-aware benchmark systems. We also evaluate the adapted translation distributions by looking at translation probabilities under specific topics and inspect translations of ambiguous source words.

4.6.1 Analysis of bilingual topic models

We experimented with 10, 20 and 50 topics for phrasal LDA. The diagrams in Figure 4.6 shows blocks of training and test documents in each of the three domains for a model with 20 topics. Darker shading means that documents have a higher proportion of a particular topic in their document-topic distribution. The first topic is the one affected by the asymmetric prior and inspecting its most probable phrase pairs showed that it had 'collected' a large number of stop word phrases. This explains why it is the topic that is most shared across documents and domains. There is quite a clear horizontal separation between documents of different domains, for example, topics 6, 8, 19 occur mostly in TED, NC and CC documents respectively. The overall structure is very similar between training (top) and test (bottom) documents, which shows that test inference was successful in carrying over the information learned on training documents. There is also some degree of topic sharing across domains, for example topics 4 and 15 occur in documents of all three domains. Figure 4.7 shows examples of latent topics found during inference on the training data. Topic 4 contains frequent phrase pairs related to health issues, with occurrences in all three corpora. Topic 8 and 11 seem to be about politics and economy and occur frequently in documents from the NC corpus. Topic 9 clusters phrase pairs related to sciences like physics and biology and mostly occurs in the TED corpus. Topic 14 contains phrases related to hotels and topic 19 is about web and software, both frequent themes in the CC corpus.

4.6.2 Evaluation according to BLEU

In Table 4.6 we compare our topic-adapted features when added separately to the baseline phrase table. The inclusion of each feature improves over the concatenation baseline but the combination of all four features gives the best overall results. Though the relative performance differs slightly for each domain portion in the test set, overall the

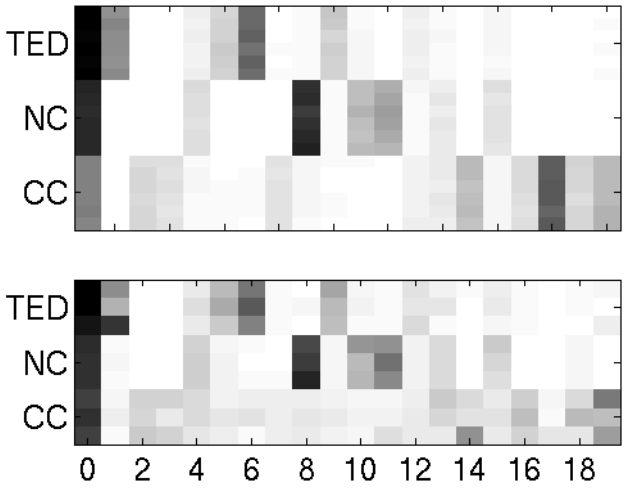


Figure 4.6: Document-topic distributions for training (top) and test (bottom) documents, grouped by domain and averaged into blocks for visualisation.

Topic 4	**Topic 8**	**Topic 9**
maladie → disease	crise → crisis	univers → universe
enfants → children	européenne → european	planète → planet
les femmes → women	politiques → political	la vie →life
risque →risk	politique → policy	terre → earth
santé → health	mondiale → global	l'univers → the universe
pauvres →poor	intérêts → interests	espèces → species
l'afrique → africa	parti → party	l'eau → water
l'inde → india	avenir → future	cellules → cells
maladies → diseases	changement → change	années → years
traitement → treatment	la chine → china	mars → mars

Topic 11	**Topic 14**	**Topic 19**
% → %	hôtel → hotel	web → web
crise → crisis	vous → you	site → site
taux → rate	ville → city	contenu → content
financière →financial	plage → beach	utiliser → use
banque → bank	chambres → rooms	fichier → file
monétaire → monetary	situé → located	logiciel → software
la croissance → growth	chambres → bedrooms	données → data
les banques → banks	quartier → district	utilisateur → user
d'intérêt → interest	capitale → capital	d'exploitation → operating
l'inflation → inflation	salon → living room	réseau → network

Figure 4.7: Frequent phrase pairs from a set of 20 learned topics.

Model	Mixed	CC	NC	TED
ALL	26.86	19.61	29.42	31.88
lex(e|f,d)	26.99	19.93	29.34	32.19
trgUnigrams	27.15	19.90	29.54	32.50
docSim	27.22	20.11	29.63	32.40
P(e|f,d)	27.31	20.23	29.52	32.58
All features	**27.67**	**20.40**	**30.04**	**33.08**
>ALL	+0.81	+0.79	+0.62	+1.20

Table 4.6: BLEU scores of pLDA features (50 topics), separately and combined.

adapted lexical weight is the weakest feature and the adapted translation probability is the strongest feature. We also performed feature ablation tests and found that no combination of features was superior to combining all four features. This confirms that the gains of each feature lead to additive improvements in the combined model. Appendix A, Table A.1 provides METEOR scores for these experiments, according to which the features fall into two groups: 1) *lex* and *trgUnigrams*, 2) *docSim* and $P(e|f,d)$. While all features improve over the baseline system, the second group outperforms the first group and the combination of all features performs best, which confirms the result according to BLEU.

As mentioned in Section 4.5.2, the higher performance of the ALL model over the IN model for Commoncrawl indicates that noise in the parallel data is leveled out by adding data from other domains. Therefore, the good performance of the topic-adapted model on CC data also shows that the approach is able to deal with noisy data. By grouping translations of source phrases into topic clusters, the resulting probability distributions are more peaked and therefore more robust against noise. Unless a bad translation was seen several times in a similar context, it is less likely that noisy translations will be chosen in a given topical context.

In Table 4.7 we compare topic-adapted models with varying numbers of topics to the concatenation baseline. We see a consistent gain on all domains when increasing the number of topics from three to five and ten topics. This is evidence that the number of domain labels is in fact smaller than the number of underlying topics. The optimal number of latent topics varies for each domain and reflects our insights from section 4.5.2. The CC domain was shown to be the most diverse and the best performance on

Model	Mixed	CC	NC	TED
ALL	26.86	19.61	29.42	31.88
3 topics	26.95	19.83	29.46	32.02
4 topics	*27.22	20.00	29.83	32.27
5 topics	*27.48	19.98	29.94	33.04
10 topics	*27.65	20.34	29.99	**33.14**
20 topics	*27.63	20.39	29.93	33.09
50 topics	***27.67**	20.40	**30.04**	33.08
100 topics	*27.65	**20.54**	30.00	32.90
>ALL	+0.81	+0.93	+0.62	+1.26

Table 4.7: BLEU scores of baseline and topic-adapted systems (pLDA) with all 4 features and largest improvements over baseline.

the CC portion of the test set is achieved with 100 topics. Likewise, the TED domain was shown to be least diverse and here the best performance is achieved with only 10 topics. The best performance on the entire test set is achieved with 50 topics, which is also the optimal number of topics for the NC domain. The bottom row of the table indicates the relative improvement of the best topic-adapted model per domain over the ALL model. Using all four topic-adapted features yields an improvement of 0.81 BLEU on the mixed test set. The highest improvement on a given domain is achieved for TED with an increase of 1.26 BLEU. The smallest improvement is measured on the NC domain. This is in line with the observation that distributions in the NC in-domain table are most similar to the ALL table, therefore we would expect the smallest improvement for domain or topic adaptation. We use bootstrap resampling (Koehn, 2004b) to measure significance on the mixed test set and mark all statistically significant results compared to the respective baselines with asterisk (*: $p \leq 0.01$). Appendix A, Table A.2 provides METEOR scores for the baseline system and the adapted system with different numbers of topics. While the improvements are slightly smaller than the improvements according to BLEU, they are still consistent across all portions of the test set and the same overall trend emerges as for the BLEU results.

In order to demonstrate the benefit of topic adaptation over more standard domain adaptation approaches for a diverse data set, we show the performance of two state-of-the-art domain-adapted systems in Table 4.8. Both FILLUP and LIN-TM improve over

Data	Mixed	CC	NC	TED
FILLUP	27.12	19.36	29.78	32.71
LIN-TM	27.24	19.61	29.87	32.73
pLDA	***27.67**	**20.40**	**30.04**	**33.08**
>FILLUP	+0.55	+1.04	+0.26	+0.37
>LIN-TM	+0.43	+0.79	+0.17	+0.35

Table 4.8: Comparison of best pLDA system with two domain-aware benchmark systems, according to BLEU.

the ALL model on the mixed test set, by 0.26 and 0.38 BLEU respectively. The largest improvement is on TED while on the CC domain, FILLUP decreases in performance and LIN-TM yields no improvement either. This shows that relying on in-domain distributions for adaptation to a noisy and diverse domain like CC is problematic. The pLDA model yields the largest improvement over the domain-adapted systems on the CC test set, with in increase of 1.04 BLEU over FILLUP and 0.79 over LIN-TM. The improvements on the other two domains are smaller but consistent. Appendix A, Table A.3 shows the METEOR scores for the same experiments. As before, the absolute improvements in METOR are smaller than in BLEU but the metrics still ranks the pLDA model consistently higher than the domain-adapted models.

We also compare the best model from Table 4.7 to all other models in combination with linearly interpolated language models (LIN-LM), interpolated separately for each domain. The results are shown in Table 4.9 (BLEU) and Appendix A, Table A.4 (METEOR). Though the improvements are slightly smaller than without adapted language models, there is still a gain over the concatenation baseline of 0.68 BLEU on the mixed test set and similar improvements to before over the benchmarks (on TED the improvements are actually even larger). Thus, we have shown that topic-adaptation is effective on test sets of diverse documents and that we can achieve substantial improvements even in comparison with domain-adapted translation and language models.

4.6.3 Properties of adapted distributions

The first column of Table 4.10 shows the average entropy of phrase table entries in the adapted models according to $P(t|s,d)$[10] versus the all-domain model, computed over

[10]The additional adapted features are not probabilistic.

Model	Mixed	CC	NC	TED
LIN-LM				
+ ALL	27.16	19.71	29.77	32.46
+ FILLUP	27.20	19.37	29.84	32.90
+ LIN-TM	27.34	19.59	29.92	33.02
+ pLDA	***27.84**	**20.48**	**30.03**	**33.57**
>ALL	+0.68	+0.77	+0.26	+1.11

Table 4.9: Combination of all models with additional LM adaptation (pLDA: 50 topics), according to BLEU.

Set	Model	Avg entropy	Avg perplexity
CC	pLDA	**3.74**	**9.21**
	ALL	3.99	10.13
NC	pLDA	**3.42**	**6.96**
	ALL	3.82	7.51
TED	pLDA	**3.33**	**9.17**
	ALL	4.00	9.71

Table 4.10: Average entropy of translation distributions and test set perplexity of the adapted model.

source tokens in the test set that are content words. Specifically, it was computed by averaging the entropies of all phrase table entries associated with each source content token in the test set. The entropy decreases in the adapted tables in all cases which is an indicator that the distributions over translations of content words have become more peaked. The second column shows the average perplexity of target tokens in the test set which is a measure of how likely a model is to produce words in the reference translation. We use the alignment information between source and reference and therefore limit our analysis to pairs of aligned words, but nevertheless this shows that the adapted translation distributions model the test set distributions better than the baseline model. Therefore, the adapted distributions are not just more peaked but also more often peaked towards the correct translation.

Table 4.11 shows examples of ambiguous French words that have different preferred translations depending on the latent topic. The word *régime* can be translated as

diet, *regime*, *rule* and *restrictions* and the model has learned that the probability over translations changes when moving from one topic to another (preferred translations under the ALL model are marked with *, multiple starred translation for the same source word have equal probability). For example, the translation to *diet* is most probable under topics 4 and 6 and the translation to *regime* (which would occur in a political context) is most probable under topic 8. Topic 6 is most prominent among TED documents while topic 8 is found most frequently in News Commentary documents which have a high percentage of politically related text. The French word *répertoire* translates most frequently to *repertoire* and *directory*, and the latter translation is the preferred translation in topic 19 which clusters IT-related phrase pairs and is frequent in the Commoncrawl corpus. The French word *noyau* can be translated to *nucleus* (physics), *core* (generic) and *kernel* (IT) among other translations and the topics that exhibit these preferred translations (topics 9, 11 and 19) can be attributed to TED (which contains many talks about science), NC and CC (which contains many documents about IT). The last example, *démon*, has three frequent translations in English: *devil*, *demon* and *daemon*. The translation as *daemon* refers to a computer process and would occur in IT-related documents. The topic-phrase probabilities reveal that its mostly likely translation as *daemon* occurs under topic 19, which as mentioned before clusters IT-related phrase pairs. These examples show that our model can disambiguate phrase translations using latent topics.

4.6.4 Examples of topic-specific translations

As another motivating example, in Figure 4.8 we compare the output of our adapted models to the output produced by the all-domain baseline for the domain-relevant word *noyau* from Table 4.11. The example sentences are taken from test documents of the TED corpus, News Commentary corpus and Commoncrawl corpus, respectively[11]. While the ALL baseline translates each instance of *noyau* to *nucleus*, the adapted model translates each instance differently depending on the inferred topic mixtures for each document and always matches the reference translation. The probabilities in brackets show that the chosen translations were indeed the most likely under the respective adapted model. While the ALL model has a flat distribution over possible translations, the adapted models are peaked towards the correct translation. This shows that topic-

[11]The document topic distributions of all examples in this section (for a model with 20 topics) are shown in Appendix A, Figure A.1. Note that the adapted features are computed using the full posterior distribution of which the document topic distribution is one factor.

régime		
topic 4	diet = 0.49	plan = 0.15
topic 6	diet = 0.79	diet aids = 0.04
topic 8	regime* = 0.82	rule = 0.05
topic 15	rule = 0.45	regime's = 0.16
topic 19	restrictions = 0.53	diplomats = 0.10
répertoire		
topic 10	repertoire = 1.00	n/a
topic 18	repertoire = 0.67	repertory = 0.14
topic 19	directory* = 0.66	folder = 0.12
noyau		
topic 9	nucleus* = 0.89	core = 0.01
topic 11	core* = 0.93	inner = 0.03
topic 19	kernel = 0.58	core = 0.11
démon		
topic 6	devil = 0.89	demon = 0.07
topic 8	demon* = 0.98	devil = 0.01
topic 19	daemon = 0.95	demon = 0.04

Table 4.11: The two most probable translations of the French source words *régime*, *répertoire*,*noyau* and *démon* and their probabilities under different latent topics (*: preferred by ALL). Some representative topics according to manual inspection are 4: health issues, 8: politics, 9: science, 11: economy, 19: IT.

specific translation probabilities are necessary when the translation of a word shifts between topics or domains and that peaked, adapted distributions can lead to more correct translations.

Figure 4.9 shows some more examples where the adapted translation model corrects the wrong lexical choice of the baseline model, here for the ambiguous words *flux* and *altération*[12]. In both cases, the distributions over translations are also more peaked in the adapted model. In the first example, *flux* is translated wrongly to *flow* by the ALL baseline and translated to *stream* by the pLDA model, which is the more appropriate translation in this technical context. In the second example, the context requires a translation of *altération* in the medical sense which is *impairment*, but the baseline model has chosen the more general translation *alteration*.

[12]The topic-specific translations for these examples can be found in Appendix A, Table A.5.

Src:	"il suffit d'éjecter le *noyau* et d'en insérer un autre, comme ce qu'on fait pour le clônage."
BL:	"it is the **nucleus** eject and insert another, like what we do to the clônage."
pLDA:	"he just eject the **nucleus** and insert another, like what we do to the clônage." (nucleus = **0.77**)
Ref:	"you can just pop out the **nucleus** and pop in another one, and that's what you've all heard about with cloning."
Src:	"pourtant ceci obligerait les contribuables des pays de ce *noyau* à fournir du capital au sud"
BL:	"but this would force western taxpayers to provide the nucleus of capital in the south"
pLDA:	"but this would force western taxpayers to provide the **core** of capital in the south" (core = **0.78**)
Ref:	"but this would unfairly force taxpayers in the **core** countries to provide capital to the south"
Src:	"le *noyau* contient de nombreux pilotes, afin de fonctionner chez la plupart des utilisateurs."
BL:	"the nucleus contains many drivers, in order to work for most users."
pLDA:	"the **kernel** contains many drivers, to work for most users." (kernel = **0.53**)
Ref:	"the precompiled **kernel** includes a lot of drivers, in order to work for most users."

Figure 4.8: The pLDA model yields correct translation of *noyau* for test docs from TED, NC and CC (adapted translation probabilities are shown in brackets). The baseline translation probabilities are: nucleus = 0.27, core = 0.27, kernel = 0.23.

Src:	"cela permet d'afficher le *flux* que vous êtes en train de diffuser."
BL:	"this allows you to display the flow that you're trying to disseminate." (flow: 0.25 stream: 0.20)
pLDA:	"this allows you to display the **stream** that you're actually stream." (stream: **0.43** flow: 0.01)
Ref:	"this allows to display the **stream** you are actually streaming."
Src:	"d'un point de vue médical, mes jambes, la chirurgie laser pour l'*altération* visuelle"
BL:	"a medical point of view, my legs, the visual alteration laser to surgery" (alteration: 0.33 impairment: 0.33)
pLDA:	"a medical point of view, my legs, laser surgery to the visual **impairment**" (impairment: **0.66** alteration: 0.18)
Ref:	"from a medical standpoint, my legs, laser surgery for vision **impairment**"

Figure 4.9: The pLDA model corrects the lexical choice in translation for two ambiguous source words, *flux* and *altération*. Translation probabilities under the baseline and under the adapted model are shown in brackets.

Style versus meaning Chen et al. (2013b) observe in the output of their domain-adapted model that often the correct translation does not actually have a different meaning but is rather a more suitable translation in the given context according to style and genre. While we also notice many cases where translations change that are not topically relevant, e.g. translations of adverbs such as *mostly* and *mainly* in En-

Src:	"et que ce soit le penan dans les forêts du bornéo , ou les *acolytes* voodoo à haïti"
BL:	"and whether the penan in the forests of borneo, or the **acolytes** voodoo to haiti" (acolytes: 0.429 cheerleaders: 0.143)
pLDA:	"and whether the penan in the forests of the borneo , or the cheerleaders voodoo to haiti" (acolytes: 0.204 cheerleaders: 0.151)
Ref:	"and whether it is the penan in the forests of borneo , or the voodoo **acolytes** in haiti"
Src:	".. indice de corruption .. relatif à la *propension* des entreprises à payer des pots-de-vin à l'étranger."
BL:	".. corruption index .. relative to the corporate **propensity** to pay bribes abroad." (propensity: 0.345 willingness: 0.172)
pLDA:	".. corruption index .. relative to the willingness of companies to pay bribes abroad." (propensity: 0.198 willingness: 0.172)
Ref:	".. bribe payers index .. in terms of the **propensity** of companies to pay bribes overseas."

Figure 4.10: Example where the pLDA model did not yield the correct translation due to flat topic mixtures, resulting in unreliable adapted probabilities.

glish, the above examples show that the topics do capture the translation differences of polysemous words. Often, the documents in which these topics are expressed also vary in terms of style and genre, for example, many documents in the Commoncrawl corpus are much more informal than the articles from the News Commentary corpus. Therefore, the topics that capture the content of these different documents will also capture the difference in style to some extent. It would be possible to model topical and stylistic dimensions separately, which is the scope of the work on language model adaptation by Hsu and Glass (2006).

Effect of flat topic distributions In Figure 4.10, we show two examples where the adapted model does not outperform the baseline. In these examples, both the baseline and the adapted model have a preference for the correct translations (*acolytes* $\rightarrow$ *acolytes*, *propension* $\rightarrow$ *propensity*), but the adapted model chose an incorrect translation instead (*cheerleaders*, *willingness*)[13]. We believe this has to do with the relative peakedness of the inferred topic mixtures. It is a measure of how well the model will be able to disambiguate source phrases and therefore how reliable the adapted translation distributions will be. Because our model includes prior distributions, even a completely uniform topic distribution would not recover the baseline probabilities. Hence, there is no built-in backoff to the baseline distribution when topic inference re-

[13]The topic-specific translations for these examples can be found in Appendix A, Table A.6.

noyau	ALL	IN		pLDA
nucleus	0.270	TED:	0.647	0.769
core	0.270	NC:	0.853	0.779
kernel	0.233	CC:	0.389	0.525

Table 4.12: Translation probabilities of French word *noyau* for all-domain, in-domain and topic-adapted models. Note that for the in-domain and topic-adapted models the probabilities come from different phrase tables.

turns inconclusive results for a given document. For both examples in Figure 4.10, we observed flat topic mixtures where a lot of mass was assigned to topic 0 and none of the remaining topics accumulated more than ~ 0.16 of the mass. As we can see from the probabilities for the correct and wrong translation under each model in brackets, the distribution over translations inferred by pLDA is flatter than the baseline distribution, making the choice of a wrong translation more likely.

4.6.5 Recovering domains

In this section, we show for a specific domain-relevant source word that our adapted model is able to recover the preference translation for each of the three domains. Table 4.12 shows the translation probabilities for the French word *noyau* for each domain under the ALL model, under an in-domain model where the respective translation is most probable, and under the adapted model given a test document from each domain. For example, *nucleus* is the preferred translation given a TED in-domain model ($P(e|f) = 0.647$) and it is also the preferred translation under the adapted model for one of the TED test documents with $P(e|f) = 0.769$. In contrast, the ALL model has a flat distribution over the possible translations.

Table 4.13 shows the translation probabilities for the adapted models (each for a particular test document where the given translation was the correct translation) for all numbers of topics reported in Table 4.7. In all cases except the one marked in red the probability of the correct translation in the given document context was the one with the highest probability under the pLDA model. However, there is some variation in how peaked the distributions are towards the correct translation. Though there is no linear increase in peakedness from left to right, on the whole the models with larger numbers of topics seem to produce more peaked translation distributions. Even though

noyau	pLDA-3	pLDA-4	pLDA-5	pLDA-10	pLDA-20	pLDA-50	pLDA-100
nucleus	0.404	0.213	0.374	0.438	0.769	0.542	0.534
core	0.340	0.375	0.365	0.651	0.779	0.595	0.722
kernel	0.359	0.407	0.447	0.505	0.525	0.566	0.630

Table 4.13: Translation probabilities of French word *noyau* (adapted for three different test contexts) for models with different numbers of latent topics.

Mixed	CC	NC	TED
0.963	0.947	0.987	0.953

Table 4.14: Length ratio of test output for pLDA model with 50 topics.

we cannot draw conclusions from this one example, it is an indicator that those models have learned a better latent topic structure and it is a potential explanation of the better results in terms of BLEU that we saw in section 4.6.2.

4.6.6 Length ratios of test documents

As mentioned in section 4.5.1, our model is tuned on a mixed development set containing documents from all three corpora used for training and testing. This is done in order to provide sufficiently varied data for the model to learn weights that are generally applicable, independent of the structure of a specific test document. However, we noticed that the length ratio of the test output varies for the three domains as shown in Table 4.14. Obviously, in our settings we cannot tune feature weights towards a specific domain. However, it might be possible to tune optimal feature weights for each topic and use a separate topic-dependent mixture of feature weights for each test document. This could be achieved, for example, by clustering the dev set according to topics and opimising feature weights for each cluster though this would lead to small development sets for each cluster. One way to overcome this problem could be to take the average of the topic-specific feature weights and a set of global feature weights optimised on the entire development set (as is done currently) as the final weights for decoding. An alternative approach would be to tune features on the entire development set for each topic, but making the weight updates dependent on the topic proportions for each example sentence pair. That is, for a given topic, the feature optimisation

Features	ALL	Add-1	Add-1	Add-2	Add-3	Add-4	Add-1	Add-1
Baseline								
P(f|e)	0.038	0.041	0.043	0.043	0.041	0.041	0.043	0.031
lex(f|e)	0.057	0.060	0.049	0.050	0.040	0.037	0.048	0.059
P(e|f)	0.098	**-0.028**	0.061	**-0.028**	**-0.007**	**-0.013**	0.086	0.077
lex(e|f)	0.018	0.010	**-0.018**	**-0.021**	**-0.017**	**-0.019**	0.026	0.011
phrPenalty	0.159	0.134	0.159	0.124	0.099	0.124	0.124	0.107
Adapted								
P(e|f,d)	-	0.099	-	0.092	0.068	0.050	-	-
lex(e|f,d)	-	-	0.055	0.057	0.061	0.044	-	-
trgUnigrams	-	-	-	-	0.070	0.058	0.139	-
docSim	-	-	-	-	-	0.108	-	0.183

Table 4.15: Tuned translation table feature weights (average over 3 tuning runs). The adapted features (bottom of the table) are gradually added to the baseline model (ALL). Inactive features are marked with a dash.

would be biased towards example sentences whose topic mixture has a large proportion of the given topic.

4.6.7 A note on tuned feature weights

We have noticed in many experiments that when including the topic-adapted features, the weights of the corresponding baseline features decrease or go below zero. This is a strong indicator that the optimiser learns that the adapted features are more informative in predicting correct translations than the baseline features. The features that receive very small or negative weights either do not contribute to the translation choice of the model or are seen as providing opposing information to the features with positive weights. In our case, if the forward translation feature of the baseline model receives a negative weight while our adapted translation probability receives a positive weight, this confirms that the adapted feature prefers translations which are very different from what the baseline model regards as a good translation.

Table 4.15 provides some concrete examples of the tuned feature weights for different numbers of additional, adapted features. On the left side of the table, no matter how many features are added to the model, if the added feature has a counterpart in the

Model	Mixed	CC	NC	TED
ALL	26.86	19.61	29.42	31.88
Add-4	27.60	**20.57**	29.86	32.89
Add-2, replace-2	**27.67**	20.40	**30.04**	**33.08**

Table 4.16: BLEU scores of pLDA models when keeping or replacing the corresponding baseline features.

baseline model, the weight of that feature turns negative (as marked in bold). The right side of the table shows the feature weights when adding the target unigram feature and the document similarity feature in isolation. In both cases, the baseline features retain positive weights, but the added feature receives a larger positive weight. This indicates that these features are providing meaningful additional information but do not entirely replace the standard translation features. In Table 4.16 we compare the performance of two models with 4 adapted features, where the first maintains all the baseline features and in the second the two forward translation features are replaced by their adapted counterparts. The system with fewer features in total gets a small performance gain, but the difference on the mixed test set is too small to conclude that there is a real difference between the two systems.

4.6.8 WADE evaluation

In this section we take a closer look at the results on the TED test set using the WADE framework. Since out of the three test domains, the TED portion of the test set saw the largest improvements when considering both domain and topic adaptation, we want to look at the performance of each system on specific subsets of source words. Table 4.17 shows the percentage of correct words out of all aligned word pairs in the source and reference sentences for the baseline system, as well as the improvements when adding domain or topic adaptation. We distinguish output words according to whether they are content or function words[14] and whether the distributions over target words have high or low entropy[15]. The latter is an indicator of how ambiguous the distributions

[14]We consider nouns, verbs, adjectives and adverbs as content words. In addition, we exclude the paradigm of some very frequent French words from the content words and count them as function words: *avoir*, *être*, *faire*. We do the same for the particles *ne* and *pas*.

[15]High versus low entropy is determined by computing the ratio between the entropy of the distribution and the entropy of a uniform distribution over the same support, in the baseline translation table. If the ratio is higher than 0.5, the words are considered to have high entropy, otherwise low entropy.

	Baseline	**+LIN-TM**	**+FILLUP**	**+LIN-LM**	**+ pLDA**
	% correct	% improvement			
Content words	50.28	0.31	0.28	0.02	**0.56**
Function words	63.51	**0.39**	**0.33**	**0.38**	0.53
High entropy words	44.81	0.34	0.24	-0.06	**0.76**
Low entropy words	68.69	**0.41**	**0.37**	**0.33**	0.43

Table 4.17: Percentage of correctly translated words with the baseline system and improvements of different models over the baseline according to WADE. Source words are grouped by different sub-classes.

are and thus whether contextual information would be expected to have an influence on lexical choice. The number of tokens and types in each sub-class can be found in Appendix A, Table A.7.

Table 4.17 shows that while all systems yield improvements on all word sub-classes (except for LIN-LM on high entropy words), there are subtle differences between the domain-adapted and topic-adapted systems. While the domain-adapted systems yield slightly larger improvements on function and low entropy words, the topic-adapted system yields a larger percentage of correct translation of content and high entropy words. This can be seen as an indicator that topic adaptation is better at using contextual information to disambiguate word senses than domain adaptation. However, further analysis at a larger scale is needed to verify this hypothesis.

4.7 Evaluation on Commoncrawl data

In this section we present experimental results on a related data set that consists only of documents from the Commoncrawl corpus. While inspecting the test results on the Commoncrawl portion of our mixed French-English data set, we noticed that some of the crawled document pairs contained very noisy translations, such as untranslated bits of text and document pairs that are comparable rather than actual translations of each other. We wanted to make sure that this noise in the test set did not affect the quality of our results and therefore set up an additional experiment on a cleaner set of Commoncrawl documents. In addition, we wanted to further compare against baseline systems that apply domain-adaptation techniques to an automatic topical clustering of the data, using monolingual topic models.

Data	Documents	Sentences
Train	21367	502445
Dev	134	3055
Test	240	6065

Table 4.18: Statistics of cleaned French-English Commoncrawl data sets.

4.7.1 Experimental setup

Data preparation In order to get a cleaner set of documents, we first removed documents crawled from particular websites that we found to provide low quality translations (especially from travel websites or websites with adult content). We used the MALLET toolkit (McCallum, 2002) to train monolingual topic models with 50 topics on both source and target side and filtered out documents by removing entire topics on both sides, given the frequent words in the topics and the following criteria: mostly words in the wrong language, mostly adult content.

In order to ensure a certain amount of topical overlap between training, development and test data, we trained a new monolingual topic model with 50 topics on the target side of the data set and grouped the documents by topic. We assigned a portion of documents from each cluster to development and test sets and the remaining documents to the training set. The main reason for doing this was an observation that data from particular kinds of websites seemed to be overrepresented in the Commoncrawl corpus. Details of the data set are shown in Table 4.18.

Topic-adapted baselines For the topic adaptation baselines, we trained 10 monolingual source side topics on the training data (again using the MALLET toolkit) and assigned each training document to a cluster according to its most prominent topic. We built separate translation tables on each cluster of training data as well as an out-of-domain model for each cluster and combined them using two domain adaptation techniques: linear mixture models with perplexity minimisation (Sennrich, 2012b) (LIN-TM) and the phrase table fillup technique (Bisazza et al., 2011) (FILLUP). This resulted in 10 adapted phrase tables for each of the two techniques, one for each topic cluster. We also built a topic-adapted baseline with linearly interpolated language models, tuned to minimise perplexity on a development set (LIN-LM). Using the trained monolingual topic model, we ran topic inference on the development and test data and

Set		C0	C1	C2	C3	C4	C5	C6	C7	C8	C9
Dev	Docs	25	10	18	3	13	14	10	8	12	21
	Sents	573	218	376	69	170	318	192	236	404	499
Test	Docs	25	16	11	15	21	34	21	21	16	60
	Sents	448	426	229	306	371	797	672	704	352	1760

Table 4.19: Number of documents and sentences in most probable topic clusters (C0-C9) of cleaned French-English Commoncrawl development and test data.

assigned each document to the most probable topic cluster. This resulted in the topic assignments shown in Table 4.19. We observe that despite our pre-selection step, the distribution of documents to clusters is still rather unbalanced.

We decided not to train models with more than 10 topics under the assumption that splitting the data set into smaller portions would lead to unreliable estimation of the cluster-specific phrase tables. Since the topic clustering also applies to the development set, the methods that optimise development set perplexity for each topic cluster (LIN-TM and LIN-LM) could suffer from too small topic-specific development sets.

An alternative way to build a topic-adapted translation model baseline would be to collect fractional phrase pair counts from the training documents according to the document-topic distributions, which would again result in one phrase table per topic. At test time, we could select the table that corresponds to the maximum topic in a test document or set mixture weights according to some distance measure (Foster and Kuhn, 2007). In comparison to the topic adaptation baselines described above, collecting fractional counts avoids the maximum operations involved in assigning documents to clusters. On the other hand, document-level topics can be inaccurate and the gain from collecting very small document-topic proportions as fractional counts is unclear. Therefore, treating only the documents with large topic proportions as in-domain data for the respective topic and interpolating with all other documents as out-of-domain data may actually provide useful smoothing. All of the methods for topic adaptation of phrase tables described in this section are impractical for large numbers of topics because they either involve building a full phrase table for each topic or rely on topic-specific development data.

Details of pLDA models The pLDA model used in this section contains the same set of 4 adapted features as used in Section 4.6. In this setup, the features were added to

the phrase table and all of the baseline features were kept. This was due to the fact that ignoring features in the phrase table during parameter tuning was less straightforward to implement for kbest-MIRA than for PRO, but we did not expect this to have a major effect on overall performance. We optimised the weights on the model with 50 topics and used the average weights of three tuning runs to decode with all pLDA models.

4.7.2 Evaluation

In this section, we evaluate the pLDA model against several baseline systems. Apart from a simple unadapted baseline we built two systems with translation model adaptation according to topic clusters (FILLUP, LIN-TM), a system with linear language model interpolation using topic clusters (LIN-LM) and two systems with combined translation model and language model adaptation (FILLUP+LIN-LM, LIN-TM+LIN-LM). We performed three tuning runs with kbest-mira (Cherry and Foster, 2012) for all systems and took the average of the tuned weights for test set decoding. We use bootstrap resampling (Koehn, 2004b) to measure significance of the BLEU scores on the mixed test set and mark all statistically significant results compared to the baseline system with asterisk (*: $p \leq 0.01$, **$p \leq 0.001$).

Comparison of baseline models Table 4.20 compares the different unadapted and adapted baseline models. The FILLUP model yields slightly higher BLEU and METEOR scores than the baseline but without a significant improvement. The LIN-TM model yields small improvements of 0.25 BLEU and 0.14 METEOR. The LIN-LM model slightly underperforms the unadapted baseline. The combination of FILLUP and LIN-LM does not yield any improvements and the combination of LIN-TM with LIN-LM performs worse than LIN-TM alone. Clearly, traditional domain adaptation methods fail to adapt to the topical structure of a diverse data set as the Commoncrawl corpus, even when being provided with source side topic information. It is unclear what is the reason for this weak performance, but we assume that either 1) the monolingual topics are not providing the right structure to capture translation ambiguities, 2) the hard assignment to topic clusters discards too much information or 3) the topic-specific development sets are too small to train reliable adapted models. The latter could explain the poor performance of the system with language model interpolation which is usually a quite reliable technique.

System	BLEU	METEOR
Baseline	25.68	24.62
FILLUP	25.74	24.70
LIN-TM	***25.92**	**24.76**
LIN-LM	*25.55*	*24.58*
FILLUP + LIN-LM	*25.57*	*24.60*
LIN-TM + LIN-LM	25.75	24.72
LIN-TM > Baseline	+0.24	+0.14

Table 4.20: Test results of the unadapted baseline model in comparison with several topic-adapted models using domain adaptation techniques (based on 10 monolingual topics). *: $p \leq 0.01$ marks significantly better BLEU scores compared to the baseline.

Comparison of pLDA model against baselines Table 4.21 shows the results of topic adaptation with phrasal LDA in comparison with the unadapted baseline and the topic-adapted mixture model (LIN-TM) which is the best-performing model in Table 4.20. An interesting result is that we see increasing improvements over the baseline models with increasing numbers of topics. For the model with 10 topics, we gain 0.40 BLEU and 0.22 METEOR over the unadapted baseline. For the model with 100 topics, we gain 0.76 BLEU and 0.39 METEOR. We also improve over the topic-adapted baseline with all models. The relative improvement of the model with 100 topics is 0.51 BLEU and 0.25 METEOR. Even though these improvements are slightly smaller than those on the mixed data set in Section 4.6, they are still very promising in comparison to the weak results of the domain adaptation approaches. As before, we get the best performance on Crommoncrawl data with 100 topics, which suggests that increasing the number of topics could improve the results further. However, computational issues currently prevent us from significantly increasing the number of topics. The two main factors are the size of the topic-phrase pair matrix (which stores fractional co-occurrence counts) and the fact that we store topic distributions for each position in the document collection which is quite memory-intensive.

4.7.3 Monolingual versus bilingual topic adaptation

The results above show that our approach to bilingual topic adaptation performs better than using monolingual topic models combined with domain adaptation techniques.

System	BLEU	METEOR
Baseline	25.68	24.62
LIN-TM	*25.92	24.76
pLDA 10 topics	**26.07	24.84
pLDA 20 topics	**26.21	24.91
pLDA 50 topics	**26.37	24.99
pLDA 100 topics	****26.43**	**25.01**
>Baseline	+0.75	+0.39
>LIN-TM	+0.51	+0.25

Table 4.21: Test results of different pLDA models (4 adapted features) compared to an unadapted and a topic-adapted baseline (LIN-TM). *: $p \leq 0.01$, **$p \leq 0.001$ marks significantly better BLEU scores compared to the baseline.

While we are not aware of any other directly related work that compares the effect of monolingual and bilingual topic models, similar observations are made in the work of Bansal et al. (2012), who perform automatic translation sense clustering for the task of automatic bilingual dictionary induction. Using K-means clustering and a set of monolingual and bilingual distributional features for each target word type, they show that while monolingual features improve performance over the baseline, the results are even better when using bilingual features. Our own findings corroborate these results. Since in both cases the task involves learning clusters of translations with similar underlying senses, it is intuitive that encoding information from both the source and target side would find better clusters. In fact, in could be seen as clustering under the constraints imposed by links between source and target tokens.

4.8 Conclusion

In this chapter, we have presented a novel bilingual topic model based on LDA and applied it to the task of translation model adaptation on a diverse French-English data set. Our model infers topic distributions over phrase pairs to compute document-specific translation probabilities and performs dynamic adaptation on test documents of unknown origin. We have shown that our model outperforms a concatenation baseline and two domain-adapted benchmark systems with BLEU gains of up to 1.26 on

domain-specific portions of the test set and 0.81 overall. The improvements persisted over baseline and benchmark systems with adapted language models. We have also shown that a combination of topic-adapted features performs better than each feature in isolation and that these gains are additive. An analysis of the domain-specific data revealed that topic adaptation compares most favourably to domain adaptation when the domain in question is rather diverse. We have further evaluated our model on a second data set extracted solely from the Commoncrawl corpus and showed that our bilingual topic modelling approach yields better performance than the combination of monolingual topic models and domain adaptation techniques.

Different from the results presented in Foster and Kuhn (2007), we have shown that dynamic translation model adaptation can outperform both the baseline model and two cross-domain adapted translation models, as well as a cross-domain adapted language model.

The structure of the proposed model entails the following advantages. Because topic adaptation is applied at the document level, integration into an SMT system is still quite efficient because phrase tables only have to be reloaded at document boundaries. For more fine-grained adaptation such as sentence-level adaptation, translation tables or at least the adapted features would have to be reloaded at each sentence boundary. Document-level adaptation has the further advantage that using information from the entire document makes topic inference quite reliable. This assumes, however, that we can trust the document boundaries in the sense that text within a document follows a coherent topical structure. We will discuss cases where this does not hold in Chapter 6.2.

Further, the model provides a natural way of adapting probabilistic translation probabilities without having to map word-level topic assignments to phrases. This also enables the model to capture topic-specific multi-word expressions that can be more informative than evaluating each word seperately. On the other hand, statistics related to phrases are always sparser than statistics related to words, which motivates us to explore an alternative model in Chapter 5 which infers topic mixtures using only words from the sentence context.

Because of its bilingual structure, the model is likely to find topic clusters that are more suitable to capture the ambiguities faced during translation. Our comparison against baseline systems that rely on monolingual topic models in section 4.7 has provided evidence that bilingual topic models are more powerful than monolingual topic models for the machine translation task.

Possible extensions of the model include tuning topic-dependent model weights as opposed to using one set of model weights for all test documents as mentioned in section 4.6.6. Another open question is related to the variation in style in addition to the variation in topic for a given document collection. It would be possible to build a more structured topic model that learns topic hierarchies where each topic consists of more formal and more colloquial subtopics, similar to the *layman* versus *technical* distinction in the hierarchical topic model of Yang et al. (2011). Another possible extension is the inclusion of a component that infers topic mixtures from less sparse sentence-level information, which is discussed in Chapter 5.

Chapter 5

Topic Adaptation with Latent Distributional Representations

In the last chapter we showed how document context can be modelled with a topic model that learns conditional dependencies between topics, source phrases and target phrases. The basic unit of this model is a phrase and therefore topic inference depends on pairs of phrases that have been seen under specific topics during training. This modelling structure is very useful for learning phrase translation probabilities because we collect the relevant counts as part of training the topic model. However, if we want to predict the topical structure of contexts that are shorter than documents it may be useful to do topic inference using smaller units than phrases.

In general, words occur more frequently in text than phrases and can therefore have richer topic co-occurrence statistics while phrases can be quite sparse. In addition, the model in Chapter 4 only considers phrases that are consistent with the bitext word alignment, which can exclude useful words from training. If we do not have document context available but only sentence context, topic inference using words as the unit could be more reliable and we would have to store fewer parameters than when dealing with phrases.

There are several reasons for modelling context at the sentence-level instead of at the document-level. First, document context is not always available. If a user types a sentence in the text box of an online translation engine, then the systen has no contextual information beyond that sentence. Another motivation is that a given document may not be a set of topically homogemeous sentences, for example because of topical drift which can occur in longer documents in particular. When document context is given but we expect topical drift within a document, it would be possible to use auto-

matic text segmentation tools to first split the document into segments and then treat each segment as a document. However, automatic text segmentation algorithms often require parameters regarding the number of segments in a document or the preferred size of the segment, which is not always an intuitive decision. Therefore, in order to keep things simple we stick to the distinction between document context and sentence context.

The model presented in this chapter aims to capture the relationship between phrase pairs and words that frequently occur in the local context of a phrase pair, that is, other words occurring in the same sentence. The model can be applied to test sentences in order to measure semantic similarity between applicable phrase pairs and the test context.

5.1 Related work on word sense disambiguation

We start by reviewing some of the literature on word sense disambiguation (WSD) which has aimed at distinguishing the different senses of a word and classifying them in a given context. A machine translation system is faced with a similar task during the lexical selection step, where it needs to choose words in the target language that preserve the sense of the source words. Most work on word sense disambiguation and related tasks such as lexical substitution and lexical similarity tasks follows the *distributional hypothesis* (Harris, 1954) which assumes that words that occur in similar contexts tend to have similar meanings. The context-dependent nature of meaning is also assumed by (Firth, 1957) who states that one "shall know a word by the company is keeps". This idea forms part of the field of distributional semantics by defining that the meaning of words can be compared by computing the similarity of representations of their respective contexts. These representations are often based on vector space models or probabilistic models that capture the process of generating words and their context words. There has also been a lot of work on developing classifiers that use features such as word collocations, part-of-speech of a word and its surrounding words, syntactic features and topic features (Agirre et al., 2005a; Joshi et al., 2005; Boyd-Graber et al., 2007). Note that in the word sense disambiguation literature the word to be disambiguated is often referred to as the *target word*, which clashes with the definition of *target word* in a machine translation context.

The work of Cai et al. (2007) is an example of a WSD system that incorporates topic features. Their model is a naive Bayes network that contains features such as part-of-

speech of neighboring words, local collocations, syntactic patterns and bag-of-words (BOW) features of the context. They motivate including topic features alongside these other features in the network by pointing out that bag-of-words features are sparse and therefore poor at representing global context[1]. To alleviate the sparsity problem, they train a monolingual topic model that clusters the words appearing in the corpus to a predefined number of topics. The topic distributions of words in the BOW are then integrated into the baseline model and yield improved performance of the network.

Li et al. (2010) perform word sense disambiguation of words in context by comparing the context to a set of sense paraphrases of the target word from WordNet. They infer topic mixtures of the sentence context and the sense paraphrases and compare them with the cosine function.

Dinu and Lapata (2010) learn distributions over word senses in the form of lower-dimensional distributional representations using topic models in order to solve lexical similarity and substitution tasks. Before computing word similarities of test words in context, each instance is contextualised using the sense distributions of context words, thereby modifying the global sense distribution of each word instance. We will adopt a similar distributional representation, but argue that our representation does not necessarily need this disambiguation step because at the level of phrase pairs the ambiguity is already much reduced.

A problem with standard word sense disambiguation data sets is that the senses can be much more fine-grained than necessary for applications like machine translation. Instead of using predefined word senses from linguistic resources like WordNet, *crosslingual word sense disambiguation* defines the sense of a word by its translation to a target language (see the description of a recent shared task by Lefever and Hoste (2013)). This approach has two advantages: 1) it does not rely on knowledge bases that are unavailable for many languages, 2) the level of granularity of a set of senses is defined by actual word usage and distinctions made during translation which can be coarser than manually assigned sense labels.

The application of WSD techniques to the task of machine translation is not new. Approaches to incorporating WSD into MT systems include Carpuat and Wu (2007c) who integrate a feature-rich WSD classifer to improve lexical selection of an SMT system. They further extend word sense disambiguation to phrase sense disambiguation and show improved performance due to the better fit with multiple possible segmenta-

[1]Note that their definition of global context is different from ours: "global features such as single words in the surrounding BOW context" Cai et al. (2007). Instead, we distinguish between local (sentence-level) and global (document-level) context.

tions in a phrase-based system (Carpuat and Wu, 2007b). Similarly, Chan et al. (2007) augment a hierarchical phrase-based MT system with a WSD classifier using local collocations, parts-of-speech and surrounding words in the source sentence to compute an additional feature for each translation rule.

5.2 Related work on vector space models for MT

There has been some fairly recent work applying vector space models to the tasks of domain adaptation and sense disambiguation for MT. Chen et al. (2013b) propose to represent each phrase pair (f,e) with an associated domain vector $v(f,e) = <w_1(f,e),\ldots,w_C(f,e)>$ where each dimension holds a tf-idf weight for the occurrence of (f,e) in each subcorpus C. The adaptation step consist of constructing a similar vector $v(dev)$ for the development set, this time with weights characterising the occurrence of all phrase pairs in the development set in each subcorpus C. The similarity between a phrase pair and the development set is encoded in a similarity feature to learn a preference for phrase pairs with a similar vector profile as the development set. However, the model has no notion of structure beyond corpus boundaries and adaptation is cross-domain as the model is adapted to the development set.

Costa-jussà and Banchs (2010) build a vector space model that captures the source context of every training sentence that a phrase pair occurred in. Given a test input sentence and an applicable phrase pair, they compare the vector space representation of the test context to the vector space representation of all training instances for this phrase pair. A similarity feature enables the decoder to give priority to phrase pairs extracted from similar contexts. Banchs and Costa-jussà (2011) extend this work by replacing the vector space representations with latent representations learned with Latent Semantic Indexing (LSI). The idea behind our proposed model is similar but instead of explicitly computing a latent representation for all contexts in the training data, we want to learn a latent distributional representation of a phrase pair that we can directly compare to a given context. Such a representation would abstract away the noise of specific context instances and hopefully be more robust. Because context words occurring with the same phrase pair are tied together in a pseudo-document associated with a phrase pair, we can use sparse priors to constrain the model to cluster context words associated with the same phrase pair into few topics.

5.3 Phrase Pair Topic Model (PPT)

Our goal is to capture the latent semantics of translation units directly instead of defining them via document-level or sentence-level topic distributions. While we want to capture information about the text sources where the translations of source phrases and words were found in the training data, we want to abstract from the lexical forms of training contexts to learn representations that generalise better to new test contexts. We also want the representation to be independent of corpus boundaries in the training data, which can be quite arbitrary or noisy. If we can capture the semantics of translation units and represent them in a compact form, this enables us at test time to favour translation units that are semantically similar to a given test context. It also avoids having to store context representations for each training instance of a phrase pair.

Following the *distributional hypothesis*, our proposed model aims to capture the relationship between *phrase pairs* and *source words* that frequently occur in the local context of a phrase pair, that is, context words occurring in the same sentence. The idea is that for a given phrase pair, the words that occur frequently in its context are indicative of the sense that is captured by the target phrase translating the source phrase.

We assume that all phrase pairs share a global set of topics and thus, during topic inference the distribution over topics for each phrase pair is induced from the latent topics of its context words in the training data. In order to learn topic distributions for each phrase pair, we represent phrase pairs as *distributional profiles* which are the input to the topic modelling algorithm which learns topic clusters over context words. We adopt the definition of a *distributional profile* by Mohammad and Hirst (2006)

> The context (or "company") of a target word is represented by its **distributional profile (DP)**, which lists the strength of association between the target and each of the lexical, syntactic, and/or semantic units that co-occur with it. [...] Commonly used units of co-occurrence with the target word are other *words*, and so we speak of the **lexical distributional profile of a word (lexical DPW)**. The co-occuring words may be all those in a predetermined window around the target, or may be restricted to those that have a certain syntactic (e.g. verb-object) or semantic (e.g. agent-theme) relation with the target word.

but use it in a more general sense as described in the following. We define a *distributional profile* as a *pseudo-document* that contains its sentence-level context words in all of its training contexts. Stop words are filtered out from the distributional profiles to reduce noise. This differs from the original definition in that we do not employ

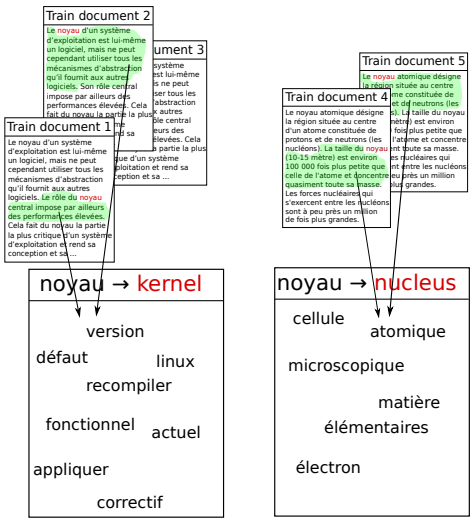


Figure 5.1: Distributional profiles extracted from the local source sentence contexts of two translation units that share the same source phrase.

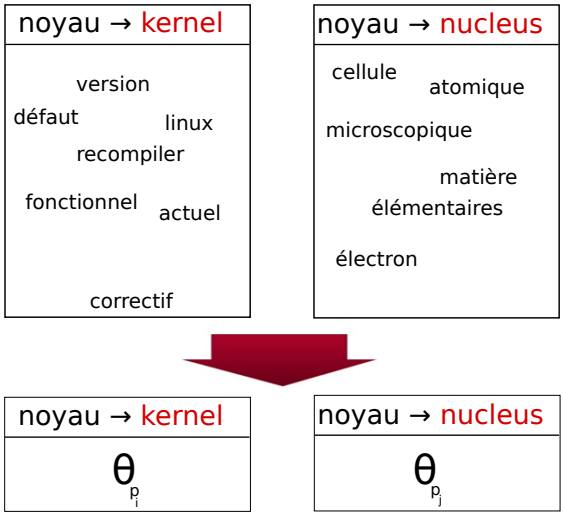


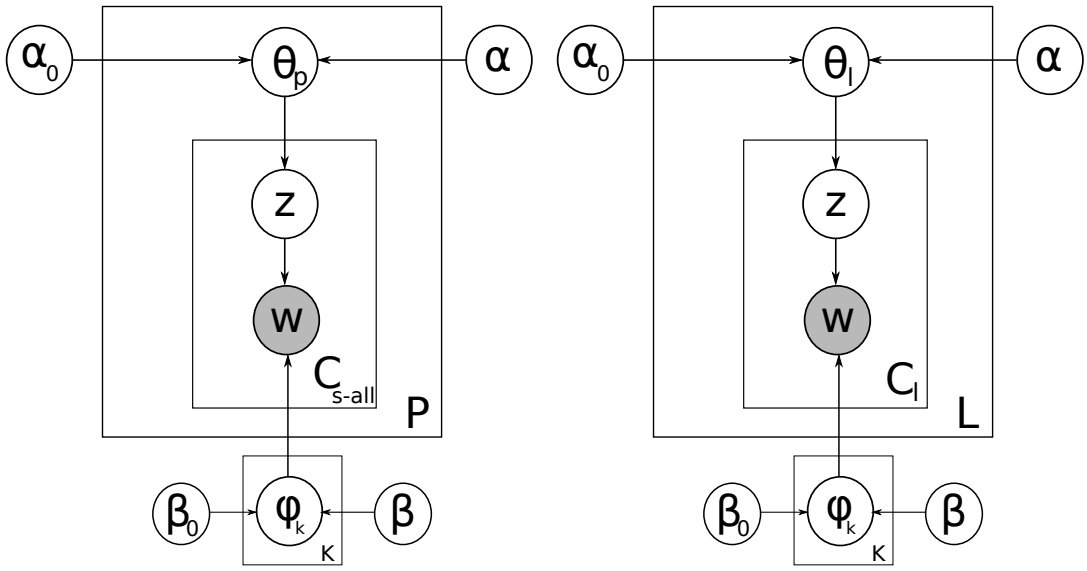
Figure 5.2: Latent topic representations derived from the distributional profiles for two translation units. All lexical information from the contexts is discarded.

measures of association strength but instead represent the raw co-occurrences of context words with phrase pairs. This idea is visualised in Figure 5.1 for two phrase pairs with the same source side but different target sides. Figure 5.2 shows the learned reduced representations that replace the lexical information from the training contexts of translation units.

Mohammad and Hirst (2006) also point out a potential problem with distributional profiles of ambiguous words

> It is clear that different senses of a word have different distributional profiles ("different company"). Using a single DP for the word will mean the union of those profiles. [...] we argue that **distributional profiles of senses or concepts (DPCs)** can be used to infer semantic properties of the senses: "You shall know a sense by the company it keeps."

Thus, a simple approach to building distributional profiles will conflate different senses of a word into the same representation. Therefore, Mohammad and Hirst propose to build separate distributional profiles for each sense of a word and showed improved performance on a word-pair ranking task. We discuss how the issue of sense conflation relates to our phrase pair DPs in Section 5.3.4.



(a) Inference on distributional profiles of phrase pairs (training).

(b) Inference on local test contexts (test).

Figure 5.3: Graphical representation of the Phrase Pair Topic Model.

5.3.1 The Generative Process

In this section we describe our model in more formal terms. Figure 5.3a shows the model for training inference on the distributional representations for each phrase pair, where C_{l-all} denotes the number of context words in all sentence contexts that the phrase pair was seen in the training data, P denotes the number of phrase pairs and K denotes the number of latent topics. The model in Figure 5.3b has the same structure but shows inference on test contexts, where C_l denotes the number of context words in the test sentence context and L denotes the number of test instances. θ_p and θ_l denote the topic distribution for a phrase pair and a test context, respectively. Figure 5.3a shows a graphical representation of the generative process for training which is described below.

For each of P *phrase pairs* pp_i in the collection

1. Draw a topic distribution from an asymmetric Dirichlet prior, $\theta_p \sim \text{Dirichlet}(\alpha_0, \alpha \ldots \alpha)$.

2. For each position c in the *distributional profile* of pp_i, draw a topic from that distribution, $z_{p,c} \sim \text{Multinomial}(\theta_p)$.

3. Conditioned on topic $z_{p,c}$, choose a context word $w_{p,c} \sim \text{Multinomial}(\phi_{z_{p,c}})$.

α and β are parameters of the Dirichlet distributions and ϕ_k denotes topic-dependent vocabularies over context words. Test contexts are generated similarly as shown in Figure 5.3b. A local test context is defined as all words in the test sentence excluding stop words, while contexts of phrase pairs in training do not include the words belonging to the source phrase. The naming in the figure refers to local test contexts L, but global test contexts can be defined similarly (see Section 6.4). The generative process for testing is described below.

For each of *L test sentences* (local) in the collection

1. Draw a topic distribution from an asymmetric Dirichlet prior, $\theta_l \sim \text{Dirichlet}(\alpha_0, \alpha \ldots \alpha)$.

2. For each position c in the *test sentence*, draw a topic from that distribution, $z_{l,c} \sim \text{Multinomial}(\theta_l)$.

3. Conditioned on topic $z_{l,c}$, choose a context word $w_{l,c} \sim \text{Multinomial}(\phi_{z_{l,c}})$.

The asymmetric prior on the topic distributions (α_0 for topic 0 and α for all other topics) as well as on the vocabulary distributions encodes the intuition that there are words occurring in the context of many phrase pairs which can be grouped under a topic with higher a priori probability than the other topics.

5.3.2 Inference in the PPT Model

As for the previous model in Chapter 4.3, we use collapsed variational Bayes (Teh et al., 2006) to infer the parameters of the PPT model. Since there is no conditional relation between source and target phrases in this model (they are modelled jointly as documents containing their context words), the posterior distribution over topics is simpler and computed as shown below

$$P(z_{p,c} = k | \mathbf{z}^{-(p,c)}, \mathbf{w_c}, p, \alpha, \beta) \propto \frac{(\mathbb{E}_{\hat{q}}[n_{.,k,w_c}^{-(p,c)}] + \beta)}{(\mathbb{E}_{\hat{q}}[n_{.,k,.}^{-(p,c)}] + W_c \cdot \beta)} \cdot (\mathbb{E}_{\hat{q}}[n_{d,k,.}^{-(p,c)}] + \alpha) \tag{5.1}$$

where $z_{p,c}$ denotes the topic at position c in the distributional profile p, $\mathbf{w_c}$ denotes all context word tokens in the collection, W_c is the total number of context words and $\mathbb{E}_{\hat{q}}$ is the expectation under the variational posterior. $n_{.,k,w_c}^{-(p,c)}$ and $n_{p,k,.}^{-(p,c)}$ are counts of topics occurring with context words and distributional profiles, respectively, and

$n_{.,k,.}^{-(p,c)}$ is a topic occurrence count. The structure of the model is similar to the standard formulation of LDA (Blei et al., 2003) except for the asymmetric prior and the diverging definition of what constitutes a document.

Training and Test Procedure Before training the topic model, we remove stop words from all distributional profiles. When inferring topics for test contexts, we ignore unseen words because they do not contribute information for topic inference. In order to speed up training inference, we limit the documents in the collection to those corresponding to phrase pairs that are needed to translate the test set. The experiments in the following two sections were carried out on the full distributional profiles, but we have experimented with reduced profiles as well.[2] Inference was run for 50 iterations on the distributional profiles for training and for 10 iterations on the test contexts. Regarding hyperparameters, we set $\alpha = 0.5$, $\alpha_0 = 2.0$, $\beta = 0.1$, β_0 = 1e-8. The output of the training inference step is a model file with all the necessary statistics to compute posterior topic distributions (which are loaded before running test inference), and the set of topic vectors for all phrase pairs. The output of test inference is the set of induced topic vectors for all test contexts.

It took $\sim$24 hours to train a model with 20 topics for 50 iterations, using 20 cores of 2.67 GHz (this includes writing model and topic mixtures files to disk several times during training which can be avoided)[3]. The training times could be reduced by improving training parallelisation. Because the distributional profiles can be of differing length, simply splitting the data into batches of equal numbers of documents is inefficient. Processors with shorter documents have to wait until processors with longer documents finish an iteration. Test inference took a total of about 6 minutes for the whole test set on a single core ($\sim$ 0.06 seconds/sentence)[4].

5.3.3 Similarity Feature

The PPT model described before learns lower-dimensional context representations for phrase pairs as well as the underlying latent topics, which are distributions over context words. This is useful because we learn a topic vector for each phrase pair from the

[2]Reducing the training contexts by scaling or sampling would be expected to speed up inference considerably. Dinu and Lapata (2010) describe that they use only the 3000 most frequent words in the corpus and scale down all word counts by a factor of 70 to speed up inference.

[3]The training time for a model with 50 topics is not directly comparable because we used only 10 parallel processors. In this setup, it took $\sim$48 hours to train a model with 50 topics for 50 iterations.

[4]We note that test inference times increased by $\sim$1 hour (or 32 seconds/document) when computing additional features at the document level as described in Chapter 7.

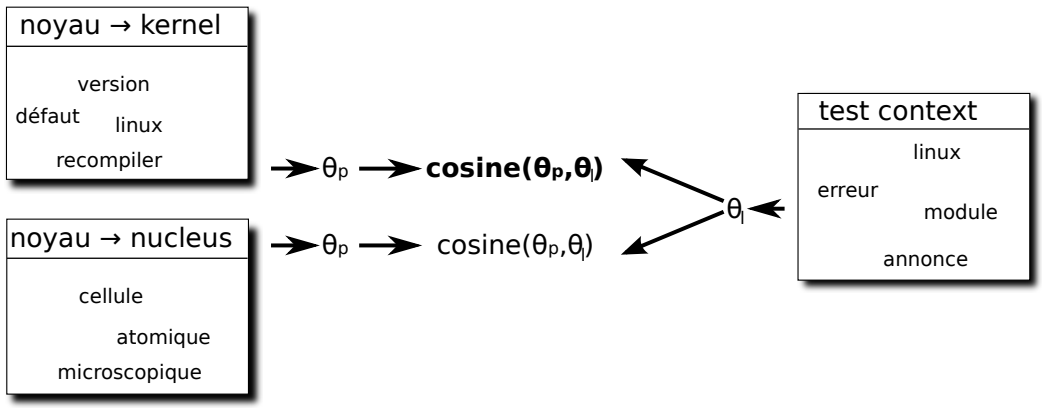


Figure 5.4: Similarity between vector representations of two applicable phrase pairs and a test context (here: French context words).

training data which captures all the information we need. Unlike the semantic feature of Banchs and Costa-jussà (2011) which we described in Section 4.4, we do not have to store topic vectors for each occurrence of a phrase pair in the training data. This is more efficient but also more appealing from a modelling point of view. The context representations can be thought of as expected latent contexts or distributions over context words in a reduced space.

Once we have learned the latent topics, we can use the model to infer topic vectors for test contexts. By comparing phrase pair topic vectors to test context topic vectors, we can evaluate how suitable a given phrase pair is to fit into a given test context from a semantic point of view. Whether the test context is a source language or target language context depends on the task and does not influence the model representation. In a standard translation task, we are given source sentences at test time and the goal is to select the most appropriate translation given the target context. Therefore, for each source phrase s we consider all applicable target phrases t_i and compute the similarities of the pairs pp_i to the test context:

$$sim(pp_i, test\ context) = cosine(\theta_{p_i}, \theta_l), \quad \forall pp_i = s \rightarrow t_i \tag{5.2}$$

We do not select the target phrase with the highest similarity score but instead provide the score as an additional feature to the translation model. The idea is visualised in Figure 5.4 for the source phrase *noyau* and two of its possible translations, *kernel* and *nucleus*. Due to context words with similar semantics, the topic vector of *noyau* → *kernel* has a larger overlap with the topic vector of the test context and will therefore more likely be chosen by the translation model.

5.3.4 Ambiguity of Phrase Pair Topic Vectors

One open question concerns the ambiguity of the topic distribution assigned to a phrase pair. Dinu and Lapata (2010) learn distributional representations for words, and in order to deal with sense ambiguity they contextualise the word representations before comparing them to other words. In the contextualisation step, every word from the local context is used to shift the sense distributions according to that words' own distribution over senses. Huang et al. (2012) follow a multi-prototype approach for constructing word embeddings to be used in word similarity tasks, arguing that "using all contexts of a homonymous or polysemous word to build a single prototype could hurt the representation, which cannot represent any one of the meanings well as it is influenced by all meanings of the word". They derive a multi-prototype representation by clustering the contexts of a given word and re-labelling each occurrence of the word with its associated context cluster. Our case is slightly different because we model pairs of phrases instead of words, which means that the ambiguity present in the source words is to some extent already resolved by the choice of words in the target phrase. However, we could think of alternatives to our current model, for example modelling target phrases or target words instead of phrase pairs which would make the model faster to train. In that case, the fact that the distributional profiles are not constrained by the source side would potentially make them more ambiguous and contextualisation could help to reduce the ambiguity before computing similarity scores.

Comparison of Distributional Profiles Figure 5.5 visualises the relationship between distributional profiles of phrases and phrase pairs. In the examples, the distributional profiles of both source and target phrase contain source words belonging to different senses of the respective phrase, for example at the top, the words occurring in the context of *noyau* belong to the senses *IT*, *science* and *generic*, while the words in the context of *kernel* belong to the senses *IT* and *food*. Thus, the monolingual profiles still contain a relatively high level of ambiguity. The distributional profile of the phrase pair *noyau* $\rightarrow$ *kernel* is the intersection of the source and target phrase contexts and preserves only the *IT* sense. This is an example of how ambiguity in the monolingual phrases can be resolved by modelling phrase pairs instead.

At the bottom of Figure 5.5, the distributional profiles for the words *noyau* and *core* as well as for the phrase pair *noyau* $\rightarrow$ *core* are shown. Here, the sense ambiguity of the source and target phrases is only partially resolved in the phrase pair which can occur in an IT context, such as "the machine has eight cores", or in an economic

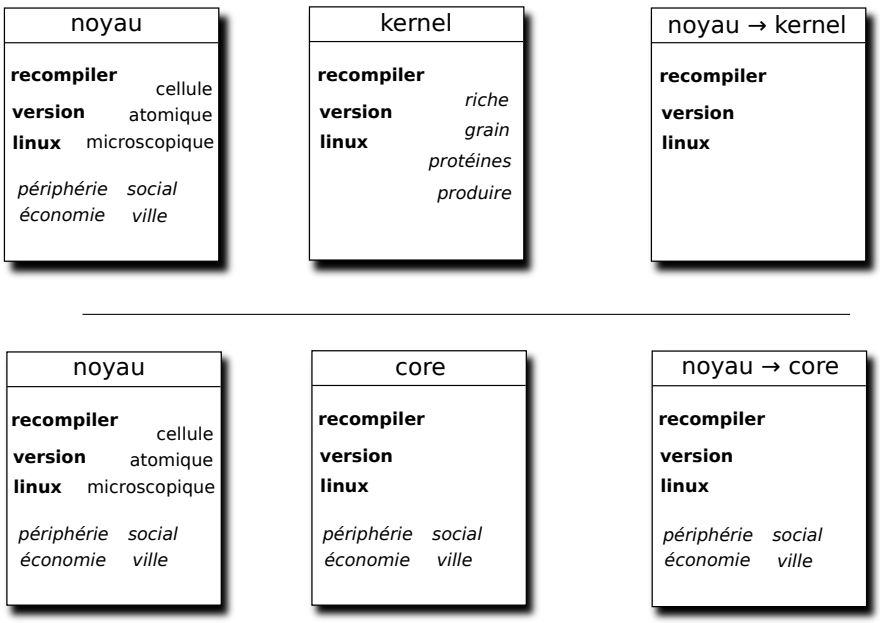


Figure 5.5: Distributional profiles for source phrase, target phrase and phrase pair (here: French context words). Top: Sense ambiguity is resolved in the phrase pair profile. Bottom: Sense ambiguity is only partially resolved in the phrase pair profile.

context, such as "the european union's core countries". In this case, contextualising the topic vector of the pair *noyau* → *core* using words in the test context would shift the distribution towards the sense that is present in the test context.

5.3.5 Comparison of similarity features to probabilistic features

One issue with the proposed PPT model in comparison to the pLDA model in Chapter 4 is that we lose the conditional formulation of target phrases given source phrases. This means that the model does not generate translation probabilities which showed to be the most discriminative of the four topic-adapted features evaluated in Chapter 4. We return to this issue in Chapter 7 where we try to derive probabilistic features from the PPT model.

A possible advantage of a similarity feature over a probabilistic feature is that the appropriateness of a phrase translation is evaluated independently for all possible phrase translations. Therefore, equally similar translations can have equally high similarity scores. However, Banchs and Costa-jussà (2011) observe that the similarity values between vectors decrease as the number of latent dimensions grows, so similarity features may be more sensitive to the number of topics than probabilistic features.

For a probabilistic feature, the presence of many likely translation options for a

Translation option	P(e|f)	P(e|f,d)	docSim	phrSim
noyau → kernel	0.233	0.525	0.929	0.968
noyau → kernel ,	0.006	0.012	0.862	0.950
noyau → kernel , you	0.006	0.012	0.862	0.950
noyau → kernel @-@	0.012	0.029	0.929	0.980
noyau → kernel with	0.006	0.016	0.887	0.852
noyau → kernel with the	0.006	0.016	0.887	0.853
noyau → core	0.270	0.108	0.909	0.518
noyau → nucleus	0.270	0.042	0.596	0.258

Table 5.1: Comparison of the probabilistic adapted feature $P(e|f,d)$ to the document similarity (*docSim*) and phrase pair similarity (*phrSim*) features. Both pLDA and PPT models were trained with 20 topics. For completeness, we also show the unadapted probabilities under $P(e|f)$ which has a flat translation distribution.

Source	Le support de reiser4 a été ajouté. Le noyau contient de nombreux pilotes, afin de fonctionner chez la plupart des utilisateurs.
Reference	It was patched in order to add support for the reiser4 file systems, and add the speakup screen reader for blind people. The precompiled kernel includes a lot of drivers, in order to work for most users.

Figure 5.6: Example of source word *noyau* and its translation in a sentence context.

given source phrase would result in a flat distribution over translations, which can have an impact when translation options covering different spans of the input compete with each other. For a similarity feature, the number of likely translations does not influence the absolute feature values of the feature and allows it to score all semantically relevant translations highly. For example, compare the feature scores for the probabilistically adapted feature (pLDA) to the document similarity (pLDA) and phrase pair similarity feature (PPT) in Table 5.1, which were adapted for the source sentence in Figure 5.6 and the document surrounding it (not shown here), respectively. In this context, the correct translation for *noyau* is *kernel*.

The probabilistic feature gives preference to the translation *kernel* ($P(e|f,d) = 0.525$) and low probabilities to all other translations.[5] Both the document and phrase pair similarity features assign high similarity scores to all target phrases that contain

[5]Note that the list of translation options was pruned for display.

the correct translation *kernel*, and lower scores to translations not containing the correct translation (here: *core* and *nucleus*). The similarity features are less precise about the concrete target phrase, but capture the semantics of similar target phrases better than the probabilistic feature. Note also that in the given example, the phrase pair similarity feature discriminates better from the semantically incorrect translations than the document similarity feature, which assigns a high similarity score to the semantically related translation *core*.

5.3.6 Qualitative evaluation of phrase pair topic distributions

In order to verify our intuition about topic distributions of phrase pairs in Section 5.3.4, we inspect the inferred distributions for three phrase pairs involving the French source phrase *noyau*: *noyau* → *kernel*, *noyau* → *nucleus* and *noyau* → *core*, for a model that was trained for the experiments in Section 5.4.1. Figure 5.7 shows the topic distributions for a PPT model with 20 topics and provides labels describing the content of some of the prominent topics. The most peaked topic distribution was learned for the phrase pair *noyau* → *kernel* and the corresponding topic is in fact about IT. The most prominent topic for the phrase pair *noyau* → *nucleus* is the science topic, though it seems to be occurring in some political contexts as well. The phrase pair *noyau* → *core* was assigned the most ambiguous topic distribution with peaks at the politics, economy and IT topics. This confirms the hypothesis from Section 5.3.4 that depending on the distributional profiles of the source and target phrases, the topical ambiguity can be resolved or preserved in the phrase pair. The phrase pair *noyau* → *core* seems to be occurring in quite diverse contexts and this ambiguity is represented in the learned topic distribution. Note also that its topic distribution overlaps with those of the other two translations, for example, like the phrase pair *noyau* → *kernel*, it can occur in IT contexts. This shows that the model captures the fact that even within a given topic there can still be ambiguity about the correct translation (both *kernel* and *core* are words that are likely to appear in an IT context).

with the caption providing labels for some of the topical peaks according to their most likely words.

Ambiguity of Phrase Pair Topic Vectors The examples in the previous paragraph show that the level of ambiguity differs between phrase pairs that constitute translations of the same source phrase. As pointed out in Section 5.3.4, introducing bilingual

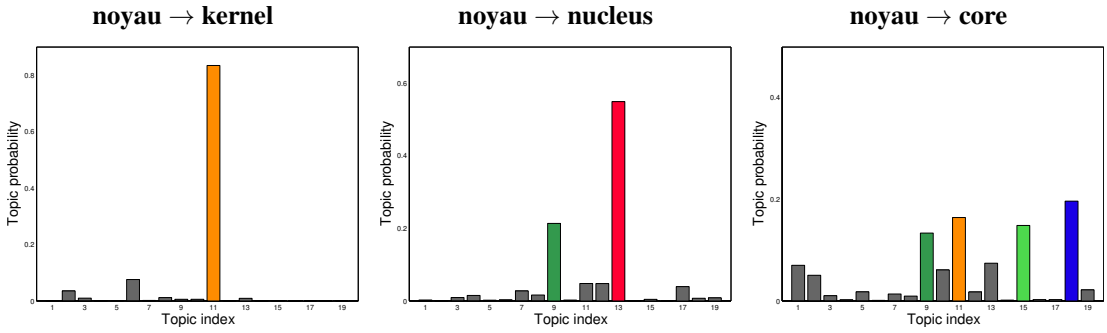


Figure 5.7: Topic distributions for source phrase *noyau* and three of its translations (20 topics without topic 0). Colored bars correspond to the topics *IT* (topic 11), *politics* (topic 9, 15), *science* (topic 13) and *economy* (topic 18) with topic proportions $\geq$10%.

information into topic modelling reduces the sense ambiguity present in monolingual text by preserving only the intersection of the senses of source and target phrases. For example, the distributional profiles of the source phrase *noyau* would contain words that belong to the senses *IT*, *politics*, *science* and *economy*, while the words in the context of the target phrase *kernel* can belong to the senses *IT* and *food* (with source context words such as *grain*, *protéines*, *produire*). Thus, the monolingual representations would still contain a relatively high level of ambiguity while the distributional profile of the phrase pair *noyau* → *kernel* preserves only the *IT* sense.

5.4 Experimental Setup and Evaluation

We evaluate the PPT model on two French-English tasks, using the data set described in Table 4.2 and reproduced here as Table 5.2.

Data	**Mixed**		**CC**	**NC**	**TED**
Train	354K	(6450)	110K	103K	140K
Dev	2453	(39)	818	817	818
Test	5664	(112)	1892	1878	1894

Table 5.2: Number of sentence pairs and documents (in brackets) in the data sets.

The first task is a regular machine translation task where we use the same setup as described in Section 4.5.1. The model is evaluated on diverse test documents from three different domains, without knowing the domain of a given document. The performance is measured in terms of case-insensitive BLEU, using the mteval-v13a.pl

script. We use the same phrase-based concatenation baseline system described in Section 4.5.1 but also compare to the pLDA model from Chapter 4 and try to extend it with our new similarity feature.

The second task is the *L2 writing assistant task* which aims at finding the translation of an L1 source phrase that best fits into a given target context and is evaluated by measuring average word accuracy of the translated L2 phrases. It provides a more intrinsic evaluation of our model because we can directly measure the model's effect on ambiguous source words and phrases. We use the same concatenation baseline to translate the French source phrases with and without integrating information from the target context. The adapted systems built on top of this baseline were not tuned for this specific task[6], but used the tuned weights of the pLDA model with only the document similarity feature included.

5.4.1 Task 1: Machine translation using source sentence context

In this section we evaluate the PPT model on a standard machine translation task. The feature weights of all models were tuned with Pairwise Ranked Optimisation (Hopkins and May, 2011) and the final results were produced using the average feature weights of three tuning runs for every setup.

PhrSim Feature and Combinations Table 5.3 shows results for the baseline system combined with an additional phrase pair similarity feature derived from the PPT model (*phrSim*) for varying numbers of topics. All models improve over the baseline and the model with 50 topics seems to do best, though there is no clear trend regarding the optimal number of topics[7]. We also ran experiments with a version of the distributional profiles where all words were stemmed, but the results were on average slightly worse than with the original profiles. Table 5.4 shows the results of a model where both the phrase pair similarity feature and the document similarity feature have been added to the baseline model. Again, all models improve over the baseline and here the model with 20 topics performs slightly better overall than the other models, but the difference is quite small. Adding both the document and the phrase pair similarity feature to the baseline yields consistently higher improvements than adding only the *phrSim*

[6]Tuning features weights for this specific task would not be difficult, but would require an additional development set of ambiguous source words in context and their translations which we currently do not have available.

[7]This is similar to the observations for the pLDA model in Chapter 4.6, where the differences only become apparent when considering the results of models with very few latent topics (3, 4 and 5 topics).

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
10 topics	27.15	19.87	29.63	32.36
20 topics	27.19	19.92	**29.76**	32.31
50 topics	**27.34**	**20.13**	29.70	**32.47**
100 topics	27.26	20.02	29.75	32.40
>Baseline	+0.48	+0.52	+0.34	+0.59

Table 5.3: BLEU scores of baseline + *phrSim* feature (6 phrase table features).

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
10 topics	27.52	20.30	29.45	**32.86**
20 topics	**27.56**	20.22	29.79	**32.86**
50 topics	27.52	**20.39**	29.78	32.77
100 topics	27.51	20.37	**29.91**	32.64

Table 5.4: BLEU scores of baseline + *docSim* + *phrSim* feature (7 phrase table features).

Model	Mixed	CC	NC	TED
pLDA	27.67	20.40	30.04	33.08
10 topics	27.54	20.31	29.88	32.93
20 topics	27.63	20.49	29.94	32.97
50 topics	27.59	20.38	29.92	32.97
100 topics	27.51	20.39	29.83	32.82

Table 5.5: BLEU scores of model with all pLDA features + *phrSim* feature (8 phrase table features) in comparison to the pLDA baseline which is slightly higher for all domains.

feature. In Table 5.5, the phrase pair similarity feature is added to a model containing all adapted pLDA features from Chapter 4. This results in a total of 8 phrase table features but does not improve the performance of the pLDA model on any of the test domains.

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
+ phrSim	**27.34**	**20.13**	**29.70**	**32.47**
+ docSim	27.22	20.11	29.63	32.40
+ phrSim	**27.52**	**20.39**	**29.78**	**32.77**
pLDA (4 adapted features)	**27.67**	20.40	**30.04**	**33.08**
+ phrSim	27.59	20.38	29.92	32.97
- P(e|f,d)	27.18	20.22	29.38	32.46
- lex(e|f,d)	27.52	**20.55**	29.64	32.92
- trgUnigrams	27.63	**20.53**	29.85	32.92
- docSim	27.43	20.41	29.60	32.79

Table 5.6: Top: BLEU scores of baseline model and added *phrSim* feature. The *phrSim* is further combined with adapted features from the pLDA model (*docSim*, $P(e|f,d)$). Bottom: model with all pLDA features + *phrSim* feature, and models where each adapted pLDA feature was removed in turn. All models were trained with 50 topics.

5.4.1.1 Analysing Feature Combinations

We proceed to analyse these feature combinations in some more detail. The top of Table 5.6 compares the performance of the phrase pair similarity feature to the simple concatenation baseline and the document similarity feature for topic models trained with 50 topics. The phrase pair similarity outperforms the baseline model and yields comparable results to the document similarity feature. Combining the document and phrase pair similarity features yields larger improvements than each of them in isolation, on all test set portions. On the TED test set, the improvement of the combined similarity features over the document similarity feature alone is 0.37 BLEU, on the mixed test set it is 0.30 BLEU. On the mixed test set, the combination of the two similarity features yields an improvement of ~0.6 BLEU over the baseline.

Combination with pLDA model The bottom part of Table 5.6 shows results of experiments where we combined all adapted features of the pLDA model with the sentence similarity feature (these results are the same as in Table 5.4 for 50 topics). Contrary to our expectations, the addition of the phrase pair similarity feature does not improve performance but instead yields a slight decrease on all domain portions of the test set (on the mixed test set, it drops by 0.08 BLEU). In order to test whether the phrase simi-

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
+ phrSim (cosine)	**27.34**	**20.13**	**29.70**	32.47
+ phrSim (JSD)	27.29	20.11	29.48	**32.49**
+ phrSim (BC)	27.10	20.01	29.35	32.17
+ docSim + phrSim (cosine)	**27.52**	**20.39**	29.78	**32.77**
+ docSim + phrSim (JSD)	27.36	20.13	**29.88**	32.54
+ docSim + phrSim (BC)	27.20	20.05	29.76	32.35

Table 5.7: BLEU scores of baseline model with added phrase pairs similarity feature using different similarity metrics (cosine, JSD, BC), and an additional document similarity feature. All models were trained with 50 topics.

larity feature overlaps or conflicts with one of the document-level adapted features, we excluded those features from the model one by one. However, it seems that none of the document-level adapted features can be replaced with the phrase pair similarity feature and removing the probabilistic feature harms performs particularly. On the Commoncrawl portion of the test set, there is a small improvement when replacing either the lexical weights or the target unigram feature with the phrase pair similarity feature, but on the other two subsets the performance is below the pLDA baseline.

Influence of Similarity Metric Next we want to investigate whether different similarity metrics affect the final translation results. Table 5.7 shows results for the phrase pair similarity feature using different similarity measures. The top of the table shows the baseline model with the addition of the phrase pair similarity feature using either cosine similarity, Jensen-Shannon Divergence (JSD) or the Bhattacharyya coefficient (BC), as well as all of these combined with the document similarity feature (using JSD as defined in Chapter 4). In both cases, the cosine metric yields the best overall results and JSD ranks second. Replacing JSD with cosine similarity for the document similarity feature did not improve results.

Tuned Feature Weights The results in Table 5.6 suggest that the document similarity and phrase pair similarity features do not encode the same information, but are at least to some extent complementary. However, combining different adapted features does not always yield gains and therefore we try to gain insight to the combined

Features	Tuned Feature Weights									
	BL	+phrS	+docS	+both	pLDA	+phrS	-P	-lex	-trgU	-docS
Baseline										
P(f|e)	0.038	0.031	0.031	0.024	0.030	0.037	0.066	0.029	0.027	0.028
lex(f|e)	0.057	0.053	0.059	0.045	0.038	0.040	0.023	0.050	0.046	0.051
P(e|f)	0.098	0.071	0.077	0.062	-	-	-	-	-	-
lex(e|f)	0.018	0.012	0.011	0.013	-	-	-	-	-	-
phrPenalty	0.159	0.152	0.107	0.160	0.109	0.158	0.184	0.186	0.185	0.141
Adapted										
P(e|f,d)	-	-	-	-	0.062	0.060	-	0.067	0.065	0.070
lex(e|f,d)	-	-	-	-	0.023	0.022	0.099	-	0.014	0.019
trgUnigrams	-	-	-	-	0.108	0.059	0.050	0.086	-	0.078
docSim	-	-	0.183	0.092	0.097	0.044	0.050	0.056	0.056	-
phrSim	-	0.070	-	0.081	0	0.048	0.088	0.092	0.073	0.035

Table 5.8: Comparison of tuned feature weights for different feature combinations. Inactive features are marked with a dash.

models by looking at their tuned feature weights[8]. Table 5.8 shows the tuned feature weights of the models in Table 5.6. First we note that when adding the *phrSim* feature on top of the baseline system, it receives a weight that is about the same as the weight for $P(e|f)$[9], which is an indicator that the feature is useful to the model. It still receives a large weight in combination with the *docSim* feature, while the weight of that feature decreases but is still large. This could either mean that the *docSim* feature is less important when the *phrSim* feature is present, or that they have a similar function and the weight is shared between them. Similarly, when we add the *phrSim* feature to the pLDA model, the weight of the *docSim* feature decreases consistently. However, if we then remove the *docSim* feature from the model (last column of Table 5.8), the *phrSim* weight does not increase which suggest that the two features do not have the same function. The *phrSim* weight does increase when either of the other pLDA features is removed from the model but the size of the weight does not correlate in any way with the observed BLEU scores. It is possible that the lack of improvement when combining all adapted features is a weakness of the optimiser and we could run experiments with different optimisers to validate this hypothesis. We explore other methods of combining document and sentence context in Chapter 6.

[8]The tuned weights are an indicator of the role of a feature in the log-linear model but have to be interpreted with care because of the complex interactions of different features.

[9]Both features range between 0 and 1.

5.4.2 Qualitative comparison of document similarity and phrase pair similarity features

discuter →	**discuss**	**chat***
docSim	**0.667**	0.488
phrSim	0.340	**0.583**

retard →	**delay**	**backwardness***
docSim	0.643	**0.661**
phrSim	0.191	**0.368**

elvis →	**the king**	**elvis***
docSim	**0.848**	0.358
phrSim	0.286	**0.355**

usages →	**customs**	**uses***
docSim	0.583	**0.619**
phrSim	0.302	**0.567**

Table 5.9: Document and phrase pair similarity scores for the translations in Figure 5.8 and Figure 5.11 (both models were trained with 50 topics). * marks correct translations in the given contexts.

The results in Table 5.6 suggested that the information from the document context and from the sentence context is to some extent complementary. Thus, in order to get a better idea about the differences of the two kinds of models in practice, we inspect some examples where their translation output differs in an interesting way.

PhrSim versus DocSim Figure 5.8 shows the output of both the *Baseline+docSim* model and the *Baseline+phrSim* model. In all four cases, the model using the *phrSim* feature selects the correct translation of the highlighted French source word (*discuter*, *elvis*, *retard*, *usages*). The contextual information suggests that the relevant topics are *IT/web* (first two examples), *economy* and *industry* and the correct translations confirm this assumption. For example, the word *chat* is very typical for an online conversation and *elvis* is the name of a text editor. Table 5.9 displays the values of the two similarity features for each of the four examples. For the first two examples (*discuter* and *elvis*), we notice that the similarity features have an opposite preference for one of the two translations, which matches with the actual translation output (*discuss* and *the king* for the document similarity feature versus *chat* and *elvis* for the phrase similarity feature). To get an idea of the meaning of these similarity values, we provide the topic vectors of the phrase pairs and the test contexts for both of these examples in Figure 5.9 and Figure 5.10, respectively. In both cases, there is a certain amount of overlap with the context for both possible translations (as indicated by colouring), but the overlap with the topic vector of the correct translation is larger.

Source	normalement, une webcam suffit pour visualiser le dispositif mais il est également parfois possible de *discuter* en ligne avec un observateur présent sur place.
DocSim	normally, a webcam is enough to visualize the device, but it is also sometimes possible to discuss online with an observer present on site.
PhrSim	normally, a webcam is enough to visualize the device, but it is also sometimes possible to **chat** online with an observer present on site.
Reference	normally a webcam gives you a view of the setup. sometimes, you can **chat** online with an observer present at the location.
Source	nous fournissons nano (un petit éditeur), vim (vi amélioré), qemacs (clone de emacs), *elvis*, joe .
DocSim	we provide nano (a little publisher), vim (vi improved), qemacs (clone of emacs), the king, joe.
PhrSim	we provide nano (a small publisher) , vim (vi improved) , qemacs (emacs,), **elvis**, joe .
Reference	nano (a lightweight editor) , vim (vi improved), qemacs (emacs clone), **elvis** and joe.
Source	on peut dire avec quelque raison qu'une grande partie du *retard* économique dans le monde s'explique par le manque de confiance mutuelle.
DocSim	we can say with some reason that much of the economic delay in the world can be explained by the lack of mutual trust.
PhrSim	we can say with some reason that much of the economic **backwardness** in the world can be explained by the lack of mutual trust.
Reference	it can be plausibly argued that much of the economic **backwardness** in the world can be explained by the lack of mutual confidence.
Source	elles auraient pu être utilisées pour des *usages* bien plus productifs tels que des investissements dans des nouvelles machines, des nouvelles usines, des maisons efficaces quant à l'énergie, ou dans la recherche.
DocSim	they could be used to customs such as much more productive investments in new machines, new factories, houses effective about energy, or in research.
PhrSim	they could be used for **uses** such as much more productive investments in new machines, new factories, houses effective about energy, or in research.
Reference	they could be used for more productive **uses** such as investments in new machinery, new factories, energy efficient houses, or research.

Figure 5.8: Translation output with *docSim* or *phrSim* feature. Here, the *phrSim* feature helps to select better translations than the *docSim* feature. Both models were trained with 50 topics.

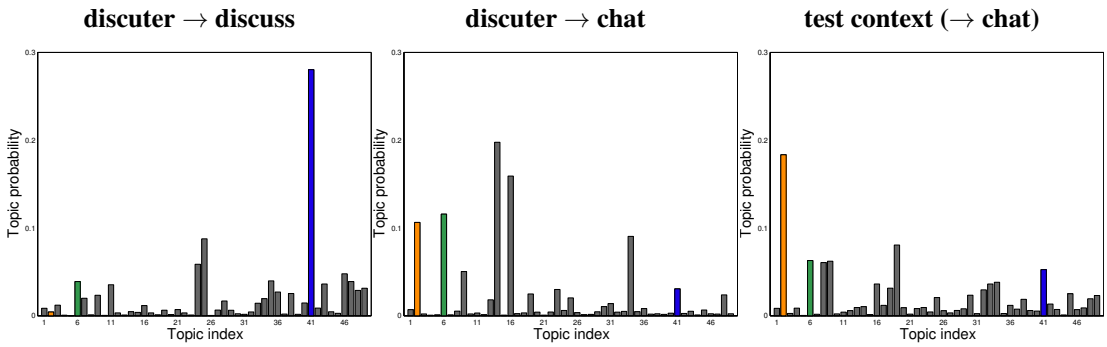


Figure 5.9: Topic mixtures of two applicable phrase pairs (left and middle) for the source word *discuter* and a given test context (right) where the correct translation is *chat*. Colored bars mark relevant overlapping topics dimensions.

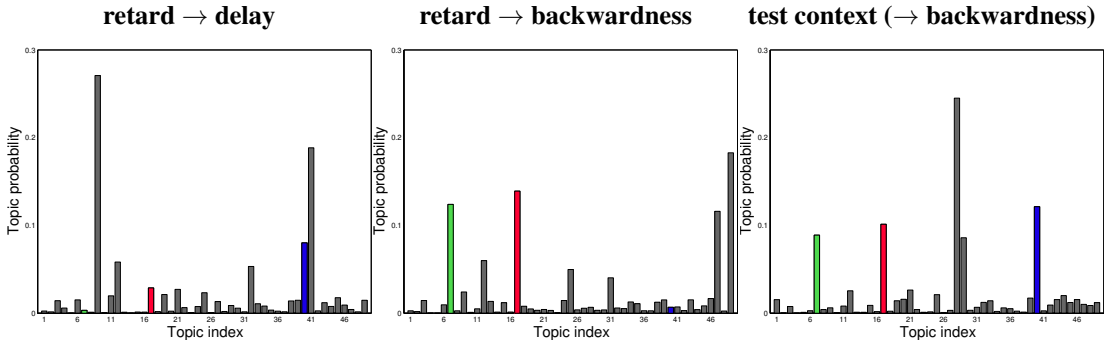


Figure 5.10: Topic mixtures of two applicable phrase pairs (left and middle) for the source word *retard* and a given test context (right) where the correct translation is *backwardness*. Colored bars mark relevant overlapping topics dimensions.

For the last two examples (*retard* and *usages*), both features have the same preference for the correct translation, but the phrase pair similarity feature separates the two translations with a larger margin (see Table 5.9). This could be one of the reasons why only the model using the phrase pair similarity feature selected the correct translation for these two words. Note that the similarity scores are not directly comparable between document similarity and phrase pair similarity because they use different similarity functions to compare the vectors (JSD vs. cosine).

PhrSim + DocSim Figure 5.11 shows the results of a model including both similarity features. Interestingly, the examples for which the two features have opposite preferences as shown in Table 5.9 are translated wrongly, while the examples where both features encode the same preference are translated correctly. This is an indicator that

DocSim+PhrSim	normally , a webcam is enough to visualize the device , but it is also sometimes possible to discuss in line with an observer present on site.
Reference	**chat**
DocSim+PhrSim	we provide nano (a small publisher), vim (vi improved), qemacs (clone of emacs), the king, joe.
Ref:	**elvis**
DocSim+PhrSim	[..] we can say with some reason that much of the economic **backwardness** in the world can be explained by the lack of mutual trust.
Reference	**backwardness**
DocSim+PhrSim	[..] they could be used for **uses** such as much more productive investments in new machines, new factories, houses effective about energy, or in research [..]
Reference	**uses**

Figure 5.11: Translation output of model using both the *docSim* and *phrSim* features, trained with 50 topics. In the first two examples, the error from the model using the *docSim* feature persists, in the last two examples, the errors are fixed by adding the *phrSim* feature. (See Figure 5.8 for input and reference sentences.)

the log-linear combination of the similarity features works well when they mutually enforce each other, but may not work that well when they provide conflicting information. In such cases, combining the similarity values in a way that favours one of the two topic mixtures could avoid that the features "cancel" each other out. For comparison, the results of the pLDA model for the same examples are shown in Figure 5.12. The pLDA model translates only one of the examples correctly, which shows that using information from the local context for topic inference can be helpful in some cases.

5.4.3 Task 2: L2 writing assistant

In this section we evaluate the PPT model on a second task in order to focus on its discriminative power in different contexts and consider the effect of using target context information on top of language model scoring. The *L2 writing assistant* task was first introduced for Semeval 2014 and is defined as follows:

> The task concerns the translation of L1 fragments, i.e words or phrases, in an L2 context. This type of translation can be applied in writing assistance systems for language learners in which users write in their target language, but are allowed to occasionally back off to their native L1 when they are uncertain of the proper word or expression in L2. These L1 frag-

pLDA	normally, a webcam is enough to visualize the device, but it is also sometimes possible to discuss in line with an observer present on site .
Reference	**chat**
pLDA	we provide nano (a small editor), vim (vi improved), qemacs (clone of emacs), the king, joe .
Reference	**elvis**
pLDA	[..] we can say with some reason that much of the economic **backwardness** in the world can be explained by the lack of mutual trust.
Reference	**backwardness**
pLDA	[..] they could be used to customs such as much more productive investments in new machines, new factories, houses effective about energy, or in research [..]
Reference	**uses**

Figure 5.12: Translation output of pLDA model trained with 50 topics. Of the four examples in Figure 5.8 that are translated correctly with the *phrSim* feature, the pLDA model translates only one example correctly. (See Figure 5.8 for input and reference sentences.)

> ments are subsequently translated, along with the L2 context, into L2 fragments.
>
> Thus, participants are asked to build a translation/writing assistance system that translates specifically marked L1 fragments, in an L2 context, to their proper L2 translation. The task find itself on the boundary of Cross-Lingual Word Sense Disambiguation and Machine Translation. Full-on machine translation typically concerns the translation of whole sentences or texts from L1 to L2. This task, in contrast, focuses on smaller fragments, side-tracking the problem of full word reordering.
> (`http://alt.qcri.org/semeval2014/task5/`)

The experiments on this task can be considered development experiments because the test instances were extracted from the same corpus that is used to train the topic model. This is because we wanted to extract a larger number of contexts with different translations for the same ambiguous source words to measure the model's ability to predict the correct translation from the context. We also tried to extract example translations that reflect different senses of the source word. Even though the model cannot directly use co-occurrence information between source words and context words (no lexical information is stored), it can use co-occurrence information between previously seen context words and topics to infer the topic mixture for the test contexts, which makes the task easier. However, the setup is useful for model development because training the topic models is considerably faster when the document collection contains

Source words	Translations	Instances
chaîne	chain: 30, string: 30, channel: 26, station: 4	90
matière	matter: 30, material: 30, subject: 10	70
domaine	domain: 30, area: 30, field: 30, realm: 23, sector: 6, estate: 5	124
état	state: 30, condition: 23	53
flux	stream: 30, flow: 30, feed: 13	73
démon	demon: 9, daemon: 8, devil: 3	20
répertoire	directory: 30, repertoire: 21, folder: 11, repertory: 5, repository: 4	71
noyau	core: 30, kernel: 30, nucleus: 30	90
régime	regime: 30, diet: 30, rule: 30	90
parties	parts: 30, parties: 30, areas: 30	90
rapport	report: 30, relationship: 19, relation: 18, reporting: 15, connection: 10	92
pièce	piece: 30, room: 30, part: 25, coin: 20, play: 20	125
programme	curriculum: 22, schedule: 5	27
accueil	hospitality: 23, welcome: 22, reception: 16	61
	Min: 2, Max: 6, Total translations: 50	1076

Table 5.10: Details of the test set extracted from a mixed domain corpus containing data from TED, News Commentary and Commoncrawl.

only the phrase pairs matching one source word per test instance. The model has also been evaluated on the real test data for that task as described in Hasler (2014).

Utility of the Task The appeal of the task setup is that translations of specific, ambiguous words and phrases are evaluated without the influence of reordering and other variations in the translation that are not important for evaluating word choice in context. The fact that the task assumes that the target context is provided (instead of the source context as in standard translation tasks) allows for a direct comparison of the impact of the language model versus wider contextual information. It is often believed that the language model is powerful enough to fix most issues of word choice because it takes an n-gram context around each target word into account. Since this n-gram context overlaps with the context bag-of-words in the L2 writing assistant task, we can compare the performance of the "translation system + context model" with and without the language model. If the addition of the context model improves even over the "translation system + language model", then the context model is likely to yield

gains in a standard machine translation setup as well. Even though in the machine translation setup the source context is given, we can expect the amount of information in the source and the (error-free) target context to be similar. Therefore, the task setup provides a useful test bed for the development of new context models.

Example Extraction Since the test data for the shared task was not available yet when this work was carried out, we evaluate the performance of the PPT model on a setup that follows the L2 writing assistant task[10]. For this purpose, we extracted a set of test instances containing 14 ambiguous French words and their English translations from the mixed domain corpus described in Section 4.5.1. Details of the test set are provided in Table 5.10. The number of translations of a given source word varies between 2 and 6 and the number of instances per source word between 20 and 120. We tried to ensure balance in the number of instances with the same translation for a given source word. Therefore, if a particular translation did not occur more frequently in the corpus, we kept the number of instances for all translations at a similar magnitude. We relied on word alignments to identify translations of source words for selecting examples, which further reduced the number of possible examples for some of the source words because of misalignments. The French words and their translations vary from being clearly separable senses of the source (homonymy) to being variations that could be interchangeable translations in some contexts (polysemy). Therefore, the difficulty in selecting the correct translation varies between source words.

Limitations Since the source phrases in our test set are all single words, the fact that they were extracted from the same data that the baseline model was trained on does not influence its translation choices. Only if the source phrases were multi-word phrases, the system would be able to memorise the phrasal context to select better translations. The only part of the baseline model that could benefit from the fact that it has seen the translations before is the language model which is what we are trying to outperform in our experiment. The entries in the reference file for this task look as follows:

```
<s id="4"> <input>voice 2 : next on the gourmet pet <f id="1">chaîne</f> ,
             decorating birthday cakes for your schnauzer .</input>
           <ref>voice 2 : next on the gourmet pet <f id="1">channel</f> ,
             decorating birthday cakes for your schnauzer .</ref> </s>
```

[10]The official test set for the shared task turned out to contain many examples which were much less ambiguous than the example set that we extracted, so we focus on the evaluation of the latter. Results on the official test set can be found in Appendix B, Table B.1 and Table B.2 or in Hasler (2014).

Here, the input is a sentence in the target language (English) and the source phrase that needs to be translated in this context is marked with the XML tag <f>. The same tag in the reference (<ref>) marks the correct translation of the source phrase. The target context is used in the PPT model to infer a sentence topic mixture. If the decoder translates the source phrase without using the language model context, then the input for the test example is simply the phrase chaîne. If the language model context is taken into account, then the target context is passed through unchanged using XML markup. This is done by defining identity translations of words or phrases using XML tags. Reordering walls in the Moses decoder (defined with the <wall/> tag) prevent the order of the context words from being changed.

5.4.3.1 Comparison of Model Performance

Table 5.11 shows the performance of a baseline model, the PPT model with 20, 50 and 100 topics and three mixture models that combine the three models with different numbers of latent topics. For comparison, we also show results of a PPT model trained with the MALLET toolkit (McCallum, 2002) instead of our own implementation. We use bootstrap resampling (Koehn, 2004b) to measure significance between each model and its respective baseline with or without language model context. Significant results are marked with asterisk (*: $p \leq 0.001$).

The baseline without context translates L1 fragments (here: source words) in isolation, without taking any advantage of the target context. The baseline with language model context uses the same translation system but passes the target context through using XML markup. Thus, the language model can score the translation in the context of the surrounding target words. Since the target context is taken from the English reference translations, it is of higher quality than we would expect from real translation output. If we wanted to make use of the target context in a standard MT setup, we could use partial target translations produced by the translation system in an incremental way, though this would require dynamically updating the context and perform incremental topic inference. In a standard MT setup, the given target contexts could be seen as "gold annotations" in the sense that an MT system would rarely produce such a perfect target context. On the other hand, this also implies that the language model can produce more reliable scores which are more difficult to beat than in a normal translation setup where the produced target context is often far from perfect.

The baseline model in Table 5.11 yields quite a low average word accuracy of 0.314, which improves to 0.726 when the language model context is taken into ac-

Topic Model	Context type	Avg word acc.	Correct	Partial	Wrong
-	No context	0.314	338	0	738
-	LM context	0.726	781	0	295
20 topics (MALLET)	L2 context	*0.485	499	47	530
	L2 + LM context	*0.789	849	1	226
20 topics	L2 context	*0.603	622	53	401
	L2 + LM context	*0.845	909	1	166
50 topics	L2 context	***0.674**	691	85	300
	L2 + LM context	***0.886**	953	1	122
100 topics	L2 context	*0.628	652	49	375
	L2 + LM context	*0.872	938	0	138
arithm-avg(20,50,100)	L2 context	*0.650	678	44	354
	L2 + LM context	*0.869	935	0	141
geom-avg(20,50,100)	L2 context	*0.670	700	43	333
	L2 + LM context	*0.883	950	0	126
minEntropy(20,50,100)	L2 context	***0.690**	712	67	297
	L2 + LM context	***0.889**	956	1	119

Table 5.11: Average word accuracy of translated L1 fragments with baseline translation model and PPT models with cosine similarity on a set of 1076 ambiguous test instances. *MinEntropy* selects the best model per test instance according to the entropy of the similarity feature. *: $p \leq 0.001$ marks significant improvement over the respective baseline with or without LM context.

count. The table also shows a breakdown into correct, partially correct and wrong translations (partially correct translations can occur because the output is not limited to single words). All models using information from the L2 context improve over the baseline model, and continue to improve even when the language model score is included. We also note that the topic models often produce partially correct translations when not using the language model. This is due to the fact that all phrase translations containing the correct translation of a source word receive high similarity scores.In combination with the language model, the target phrase that fits the immediately surrounding context better is selected.

Number of Latent Topics The comparison of topic models with different numbers of latent topics (20, 50, 100) shows that the best performance is achieved with 50 topics, though all models improve average word accuracy by more than 10% over the base-

line with language model context. In the bottom part of the table we compare models that combine the similarity values of the models with 20, 50 and 100 topics. Neither the arithmetic average nor the geometric average improve over the performance of the model with 50 topics alone. The last model combines the models by measuring the entropy of the similarity scores for a given test instance.[11] For each test instance, the model that yields the lowest entropy of the similarity scores is chosen. The intuition is that low entropy means that the similarity scores discriminate better between competing translations. The improvement of this model over the model with 50 topics is minimal, but avoids having to choose the model with the best overall performance.

Analysis by Source Word Table 5.12 shows the average word accuracies of the model with 50 topics in comparison to the baseline, broken down by the set of ambiguous French source words. For some source words, the models using L2 context and language model context translate all instances correctly (*matière*, *démon*), for other words the performance improves but is still quite weak (*régime*, *programme* and *accueil*). For the word *régime*, the likely cause is that two of the three possible translations (see Table 5.10) would be expected to occur in similar contexts (political) which makes it more difficult for the model to differentiate between the two. Similarly, the possible translations of the word *accueil* constitute different meanings but are still likely to occur in similar contexts. We also observe that there is a considerable difference in terms of the influence of the language model. In some cases, the topic model can provide most or all of the disambiguation, for example for the words *état* and *noyau*. In other cases, the language model is doing most of the work, for example for the words *matière* and *rapport*. This indicates that for some words, the crucial information to disambiguate their meaning is found in the immediate context around the word while for other words the information from the rest of the sentence is more important.

Comparison to Source Context We also conducted experiments using source context instead of target context with test examples extracted in the same fashion as before. However, the results of these experiments are not entirely conclusive (see Table 5.13). Even though the model using L1 context yields better performance when decoding without the language model context (average word accuracy = 0.633 compare to 0.603

[11] For this purpose, the similarity values are normalised to sum to 1, even though they are not normalised for their use as feature values.

Topic Model	Context Type	chaîne	matière	domaine	état	flux	démon	répertoire
-	None	0.333	0.420	0.244	0.558	0.403	0.368	0.429
-	LM	0.756	0.797	0.699	0.692	0.861	0.947	0.786
50 topics	L2	0.744	0.536	0.797	0.962	0.722	0.947	0.729
	L2+LM	0.944	1.000	0.902	0.962	0.917	1.000	0.871
Topic Model	**Context Type**	**noyau**	**régime**	**parties**	**rapport**	**pièce**	**programme**	**accueil**
-	None	0.337	0.337	0.326	0.319	0.234	0.000	0.000
-	LM	0.921	0.539	0.944	0.736	0.613	0.115	0.483
50 topics	L2	0.949	0.618	0.702	0.291	0.677	0.538	0.250
	L2+LM	0.989	0.697	0.966	0.890	0.806	0.692	0.742

Table 5.12: Average word accuracy of translated L1 fragments, broken down by French source words.

Topic Model	Context type	Avg word acc.	Correct	Partial	Wrong
20 topics	L1 context	**0.633**	653	57	366
	L1 + LM context	0.786	845	1	230
20 topics	Stemmed L1 context	0.520	535	48	493
	Stemmed L1 + LM context	0.818	880	1	195
20 topics	L2 context	0.603	622	53	401
	L2 + LM context	**0.845**	909	1	166

Table 5.13: Average word accuracy of translated L1 fragments using L1 context vs. stemmed L1 context vs. using L2 context (all models were run for 200 iterations).

for L2 context), the model with L2 context yields better results when including the language model context (average word accuracy = 0.847 compare to 0.786 for L1 context). This result is unintuitive and should be investigated further. We also tested a model that uses stemmed representations of the source context to learn a topic model. This model performs worse than the model with regular L1 context but better in combination with the language model. Again, we do not currently have a good explanation as to why the relative model performance changes when adding in a language model.

Source	the hotel is personally managed by your host, david, who will make sure you receive a true scottish <f id="1">accueil</f> and are well looked after during your stay.
Baseline	<f id="1">hospitality</f>
50 topics	<f id="1">**welcome**</f>
Reference	<f id="1">**welcome**</f>
Source	the hotel monopol combines this ideal, central location with excellent <f id="1">accueil</f> and traditional charm.
Baseline	<f id="1">home</f>
50 topics	<f id="1">**hospitality**</f>
Reference	<f id="1">**hospitality**</f>
Source	we also offer a complimentary internet terminal at the <f id="1">accueil</f> and wlan internet access in the rooms.
Baseline	<f id="1">home</f>
50 topics	<f id="1">**reception**</f>
Reference	<f id="1">**reception**</f>

Figure 5.13: Translations of L1 phrases using the baseline and an adapted system with 50 topics. Both systems were run with the language model context (+LM context).

5.4.4 Qualitative evaluation of phrase pair similarity feature

To give an intuition about the effect of the phrase similarity feature, we consider three example sentences from the *L2 writing assistant* task in Table 5.13 and the adapted translation probabilities for three translations of the ambiguous word *accueil*. Each of the examples occurs in a slightly different context and requires a different translation of the ambiguous word. As shown in Table 5.14, the feature values differ in each of the contexts and the translation with the highest similarity value is the correct translation. We also see that the similarity value of the best translation (which lies in the interval $[0, 1]$) differs quite a lot, depending on the learned topic representation for the phrase pairs and on the informativeness of the test context.

accueil $\rightarrow$	**welcome**	**hospitality**	**reception**
Sentence 1	**0.552**	0.144	0.325
Sentence 2	0.494	**0.818**	0.526
Sentence 3	0.248	0.691	**0.910**

Table 5.14: Feature values of *phrSim* feature for translations of French word *accueil* for the three sentences in Figure 5.13.

5.4.5 Potential improvements to the model

One problem with the current representation of the PPT model is sparsity. We modelled translation units directly without decomposing them into smaller units, like source and target phrases or words. An alternative that we did not try out is to model only the target phrases with their corresponding source context words and include the resulting similarity scores as an additional feature. Though the resulting topic representations would potentially be more ambiguous than the phrase pairs representations[12], the topic signal may still be strong enough to measure similarity with test contexts. It would also be possible to decompose target phrases further into target words and learn topic representations at the word level. However, the question would remain how to combine the topic representations of words that may have different importance within a phrase.

5.5 Conclusion

In this chapter, we have presented the Phrase Pair Topic Model (PPT) which is a new topic model for dynamic adaptation of MT systems that learns latent topics over *distributional profiles* of phrase pairs. For each phrase pair, the model learns a distribution over topics which can be compared to latent representations of a sentence context at test time in order to select the most appropriate phrase pairs for each given context. This model supplements the pLDA model from Chapter 4 in that it takes into account the sentence context instead of the whole document context and infers latent topics over context words rather than phrases. Because the model does not represent the conditional relationship between source and target phrases but considers them jointly, context adaptation is encoded in a vector similarity feature that compares phrase pair topic vectors and test context topic vectors at test time.

We have evaluated the PPT model on a standard machine translation task on a diverse test set as well as on a related task where L1 phrases are translated in a given L2 context. In both cases, we have shown that the model is able to leverage contextual information to improve translation quality.

Experimental results show that a combination of local (PPT) and global (pLDA) similarity features performs better than each of them separately and that their combined performance comes close to the performance of the pLDA model which includes an adapted probabilistic feature. However, we also saw evidence that combining lo-

[12]The distributional profiles would contain context words from training examples with other source phrases than the one we are interested in for a particular test context.

cal and global similarity works well in cases where both models mutually reinforce each other, while the outcome is unclear in cases where local and global context provide conflicting information. Furthermore, when combining all adapted features in the same model, the addition of the phrase similarity feature does not improve the overall performance. We therefore explore other ways of combining sentence-level and document-level context in the following chapter.

Chapter 6

A Combined Model of Local and Global Context

In the previous two chapters, we have seen instances of topic models for machine translation that take advantage of either the entire document context (Chapter 4) or the sentence context (Chapter 5). While both models yielded improvements in several setups, neither of them translates every instance of an ambiguous word or phrase correctly. Documents can be topically more heterogenous than sentences, for example because of the potentially broader thematic scope within a document. In this respect, we expect sentences to be more reliable sources of information because topic transitions are less likely within a sentence. On the other hand, sentences can be too short to be reliable for topic inference and there may not always be enough information within the same sentence to resolve lexical ambiguities. Therefore, both levels of context have potential advantages and drawbacks and combining them could be beneficial. An argument in favour of this distinction is that in choosing both the largest and smallest discourse units for topic inference we are more likely to 1) resolve contextual ambiguity when the sentence context is uninformative and 2) get more precise topic estimates when there is topical drift within a document.

In this chapter, we first review previous work that has looked at using different granularities of context for machine translation and analyse scenarios and concrete examples that require either document-level or sentence-level context. We also discuss existing models that combine local and global context outside the field of machine translation. We then present a simple extension of the model in Chapter 5 to take additional document context into account and show that it yields improvements over using just the local sentence context.

6.1 Previous work using local or global context for MT

There has been little published work focusing on the comparison between local (sentence-level) or global (document-level) topic information for machine translation. Eidelman et al. (2012) add topic-adapted lexical weights to the translation table of a Chinese→English MT system and learn associated feature weights that capture the usefulness of the most probable topic of a test instance, the second most probable topic and so on. They report experiments with 5, 10 and 20 topics with either local (LTM) or global (GTM) context and find that performance according to BLEU is equal or better with 10 local topics on two test sets and two different training corpora:

> Interestingly, the difference in translation quality between capturing document coherence in GTM and modeling purely on the sentence level is not substantial. In fact, the opposite is true, with the LTM models achieving better performance.

Overall, their models with 10 latent topics outperform models with 5 or 20 topics. It is striking though that the local model with 20 topics does not improve over the baseline and even underperforms it on the larger training corpus. This indicates that their local model is not suited to capture more fine-grained structure which could be due to sparsity caused by more topics and short contexts and the larger number of features to optimise. Even if the overall performance of the two models is similar on the small data set, this does not necessarily mean that they model the same phenomenon. They could simply be improving the translations of different test examples while yielding an overall similar amount of improvements.

Hewavitharana et al. (2013) work in a dialogue environment and experiment with incremental versus static context where incremental context is the conversational context up to the current utterance, which is updated for each new utterance, and static context includes the entire conversation. Static context corresponds to global document context while incremental context transitions from no context to local context and gradually to global context. They report their best results for 40 topics and incremental topic inference and show rank trajectories of 4 topics during a conversation to illustrate how the most prominent topic changes throughout the conversation. However, in both of their test setups (reference transcripts and ASR transcripts of conversations) the model with 20 topics using the entire conversational context performs almost equally well to the model with 40 topics and incremental context while the incremental models with 20 or 30 topics do not yield BLEU improvements (though they do improve TER

and NIST scores). One explanation could be that a more fine-grained topic model with 40 latent topics is more sensitive to changes in the topical distribution within a document and can therefore model differences that a coarse-grained topic model would not perceive. However, the gain of incremental modelling context is quite small and the results for the static models with document context seem more stable overall.

We could conclude from these results that local context is better modelled with fine-grained topic models and global context is better modelled with coarse-grained topic models, though this would not be in line with the results of Eidelman et al. (2012). Of their global models, the model with 5 topics performs worst and of their local models, the model with 20 topics performs worst. Such a conclusion may also be overinterpreting the evidence since the reported results from both works indicate that the models are not that stable across the range of either of the two variables (topic granularity and context size). Again, a possible explanation for the similar performance of the setups with 40 topics and incremental context versus 20 topics and static document context in the work of Hewavitharana et al. (2013) could be that each model performs well on different parts of the test set, resulting in an overall similar performance. Thus, a model combining information from both local/incremental and global context could perform better than each of them alone.

In summary, little previous work has examined the influence of different context sizes for MT and none has attempted to combine such information. Though the results in previous work are promising in the sense that different context sizes seem to be useful for improving MT output, they are not entirely conclusive in terms of a modelling advantage for either of the options. Therefore, we aim to address some of these questions in this chapter.

6.2 Local versus global context modelling for MT

For documents with long sentences that contain a lot of potentially topical information, it may be less important to use the entire document context. On the other hand, for documents with short sentences, like movie subtitles or other transcribed conversations, the sentence context may be a lot less informative. In that case, the only way to disambiguate an occurrence of an ambiguous word or phrase would be to inspect the surrounding context for hints about the current topic. While it is possible to set a fixed window size around each word or phrase ignoring sentence boundaries, we want

	pLDA model	PPT model
Plus		
	• modelling topics over phrase pairs can capture bilingual structure	• words occur more frequently than phrases which makes them more reliable topic indicators
	• can capture non-compositional multi-word expressions	• removing stop words and stemming do not affect search space during decoding
	• combination of several adapted features	• fast topic inference and adaptation
Minus		
	• if test context word does not have phrase table entry, it does not contribute to topic mixture	• no conditional relation between source and target phrases

Table 6.1: Comparison of the models from Chapter 4 and Chapter 5.

to avoid using different context scopes for tokens in the same sentence[1].

Advantages of document-level adaptation are that topic inference is potentially more reliable because more context is given. However, this only holds if the document can be assumed to be topically consistent. Document-level adaptation has an efficiency advantage as well, because we do not have to recompute features for every test sentence. On the other hand, sentence-level adaptation can be more precise because of the locality of the information and unlike document context, the sentence context is always available. However, the success of sentence-level adaptation depends on the length of the sentence and its actual information content.

In order to decide which model is more suitable for extensions, Table 6.1 compares the advantages and disadvantages of our two models that operate at the document-level and at the sentence-level, respectively. The outcome is rather balanced and the most imortant difference with a view to extending the models is the faster inference in the PPT model which will make it easier to take different contexts into account.

[1]This is both for practical reasons and because we prefer scopes that respect discourse boundaries.

Impact of context on translation In the following, we assume three possible relations between the information in different contexts that can affect machine translation:

1. Document and sentence context have a similar topic distribution
   → Both models are equally appropriate

2. The sentence context has a flat or ambiguous topic distribution
   → Document-level topic distribution preferred

3. Conflicting topic distributions (due to erroneous document segmentation, topical drift, more relevant information in sentence context)
   → Sentence-level topic distribution preferred

We have seen examples of case 1.) in the last chapter where both similarity features expressed a preference for the same translation, for example in Table 5.11. Figure 6.1 provides example sentences for cases 2.) and 3.) where the similarity features derived from document and sentence context do not encode the same preference. The feature values corresponding to these examples for a model with 20 topics are shown in Table 6.2, as well as a description of the peaks in topic distributions of the document (global) and sentence (local) context, respectively.

In the top example in Figure 6.1, the sentence topic mixture with peaks at both the *IT* and *science* topic is more ambiguous[2] than the document topic mixture, and as a result the *docSim* feature discriminates the correct translation *nucleus* better from the wrong translation *kernel*. For this test example, the features adapted to the document-topic mixture would be preferable. The sentence pair corresponding to the bottom example is part of a document that contains two News Commentary articles due to incorrect document segmentation[3]. As a result, the document-level topic mixture wrongly indicates the *politics* topic and gives preference to the translation *régime* → *regime*. In contrast, the sentence-level topic mixture captures the actual local context (as represented by the *health* topic) and gives preference to the translation *régime* → *diet*. For this test example, the features that are adapted to the sentence-topic mixture would be preferable.

[2]This is probably due to the words *éjecter* and *insérer* which can occur in both contexts.
[3]We used a simple rule based on article headlines to split documents.

Source	C'est également beaucoup plus simple pour des organismes eucaryotes comme nous-mêmes; il suffit d'éjecter le **noyau** et d'en insérer un autre, comme ce qu'on fait pour le clônage.
Reference	It's also simpler when you go into eukaryotes like ourselves: you can just pop out the **nucleus** and pop in another one, and that's what you've all heard about with cloning.
Source	Là encore, de nombreux facteurs corroborent le phénomène, dont un **régime** alimentaire constitué de produits frits bon marché et malsains, mais la sédentarité induite par ce temps passé devant le petite écran en est aussi l'un des aspects importants.
Reference	Again, many factors underlie this, including a **diet** of cheap, unhealthy fried foods, but the sedentary time spent in front of the tv is an important influence as well.

Figure 6.1: Examples of sentence pairs with an ambiguous sentence-topic distribution (top) and with conflicting document-topic and sentence-topic distributions (bottom). Details about the topic distributions are shown in Table 6.2.

Context	**Topical peaks**	**noyau** $\rightarrow$	**nucleus***	**kernel**
global	science: 0.46	docSim	**0.880**	**0.550**
local	science: 0.17, IT: 0.22	phrSim	0.741	0.604

Context	**Topical peaks**	**régime** $\rightarrow$	**diet***	**regime**
global	politics: 0.29	docSim	0.851	0.871
local	videos: 0.18, health: 0.17	phrSim	**0.716**	**0.281**

Table 6.2: Document and phrase pair similarity values corresponding to the examples in Figure 6.1, along with the peaks in their topic distributions. Top: *docSim* discriminates the correct translation (*) to *nucleus* better from the translation to *kernel*. Bottom: *phrSim* discriminates the correct translation to *diet* from the translation to *regime*.

6.3 Related work combining local and global context

An interesting example of a model that combines local and global information is the work of Titov and McDonald (2008) who model online reviews with a Multi-grain Topic Model. The model is based on two observations about online reviews: 1) each review has a global topic related to the place or item that the review is about, 2) each review has a number of local topics related to ratable aspects such as *cleanliness* and *location* for hotels or *sound quality* and *features* for MP3 players. The local and global topic mixtures are associated with separate topic spaces and there is an assumption that some regions within a document follow a different topical pattern than the global

topic mixture. While we assume that within a given sentence, the topic mixture could deviate from the global mixture, we do not expect local and global topic mixtures to lie in different topic spaces, but rather that the reliability of their information content differs between instances.

Another example is the neural language model of Huang et al. (2012), which learns to discriminate the next word given a word sequence in the local context and the document context. Two networks, one capturing local context and one capturing global context, are learned and their scores are added to assign a score to the following word. The global context is simply represented as the weighted average of the word embeddings of all words in the document. Their motivation is that if a word is still ambiguous in the local context, the global context could help to resolve the ambiguity. However, no alternative combination methods to the additive combination are considered.

6.4 A combined local and global context model based on the Phrase Pair Topic Model

In this section, we propose a simple extension to the PPT model to include global document context. We do not modify training inference but infer topic distributions for development and test documents both at the sentence and document level. Before computing adapted translation features, local and global topic distributions are combined into new topic vectors. Thus, the topic adaptation step for each test sentence considers both local and global context when computing phrase pair similarities.

Modelling local and global context At training time, our model has access to context words only from the local contexts of each phrase pair in their distributional profiles, that is, other words in the same source sentence as the phrase pair. This is useful for reducing noise and constraining the semantic space that the model considers for each phrase pair during training. Including context words from the entire document context surrounding each training instance would yield blown up, semantically more dispersed distributional profiles. At test time, however, we are not limited to applying the model only to the immediate surroundings of a source phrase to disambiguate its meaning. We can potentially take any size of test context into account to disambiguate the possible senses of a source phrase, but for simplicity we consider two sizes of context here which we refer to as local and global context:

Local context All words appearing in the sentence around a test source phrase, excluding stop words.

Global context All words appearing in the document around a test source phrase, excluding stop words.

Currently, the model is applied to all phrase pairs in the translation options for a test sentence but we could improve efficiency by applying it only to phrase pairs that we expect to be semantically ambiguous and use dummy feature values for all other phrase pairs.

6.5 Similarity features

We define similarity features that compare the topic vector θ_p assigned to a phrase pair[4] in training to the topic vector θ_c assigned to a test context. This is equal to the definition in Section 5.3.3 except that the context vector can now come from a combination of contexts as described below. A feature is defined for each source phrase s and all of its possible translations t_i in the phrase table:

$$sim(pp_i, test\ context) = cosine(\theta_{p_i}, \theta_c), \quad \forall pp_i = s \rightarrow t_i \tag{6.1}$$

The application of a similarity feature in a structured context is visualised in Figure 6.2. On the left, there are two applicable phrase pairs for the source phrase *noyau*, *noyau* $\rightarrow$ *kernel* and *noyau* $\rightarrow$ *nucleus*, with their distributional profiles (words belonging to the *IT* topic versus the *scientific* topic) and assigned topic vectors θ_p. The local and global test contexts are similarly represented by a document containing the context words and a resulting topic vector θ_l or θ_g. The test context vector θ_c can be one of θ_l and θ_g or a combination of both. In this example, the topic vector of *noyau* $\rightarrow$ *kernel* has a larger overlap with the topic vector of the test context and is more likely to be selected during decoding.

While in this chapter we focus on exploring vector space similarity for adaptation, mostly for computational ease, it is possible to derive probabilistic translation features from the PPT model which will be addressed in Chapter 7.

Types of similarity features We experiment with local and global phrase similarity features, *phrSim-local* and *phrSim-global*, to perform dynamic topic adaptation. These

[4]The mass of the asymmetric topic 0 defined in Section 5.3.1 is removed from the vectors and the vectors are renormalised before computing similarity features.

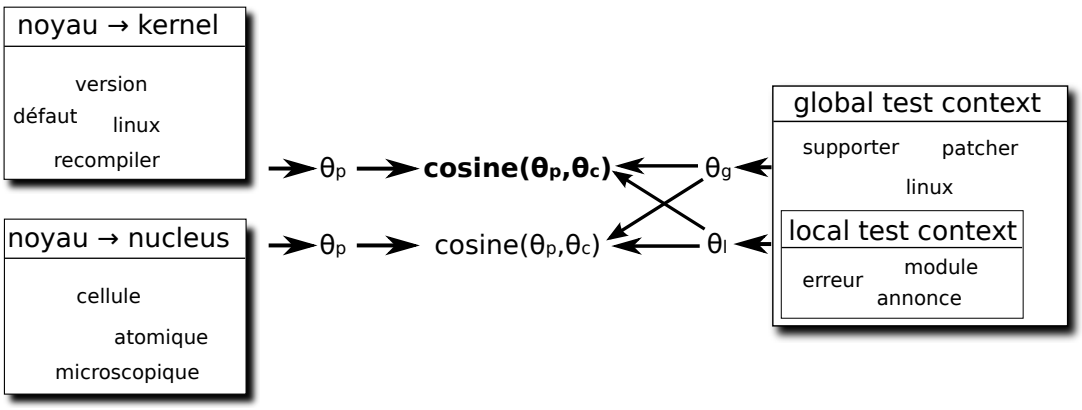


Figure 6.2: Similarity between topic vectors of two applicable phrase pairs θ_p and the topic vectors θ_l and θ_g from the local and global test context during test time.

two similarity features can be combined by adding them both to the log-linear SMT model, in which case each receive separate feature weights. Whenever we use the $+$ symbol in our results tables, the additional features were combined with existing features log-linearly. However, we also experimented with an alternative combination of local and global information where we combine the local and global topic vectors for each test context before computing similarity features.[5] We were motivated by the observation that there are cases where the local and global features have an opposite preference for one translation over another, but the log-linear combination can only learn a preference for one of the features. Combining the topic vectors allows us to potentially encode a preference for one of the contexts that depends on each test instance.

For similarity features derived from combined topic vectors, $\oplus$ denotes the additive combination of topic vectors, $\otimes$ denotes the multiplicative combination of topic vectors and $\circledast$ denotes a combination that favours the local context for longer sentences and backs off incrementally to the global context for shorter sentences. This is done by setting the interpolation weights between local and global topic vectors proportional to sentence lengths between 1 and 30 while the length of longer sentences is clipped to 30.[6] The intuition behind this combination is that if there is already sufficient evidence in the local context, the local topic mixture may be more reliable than the global mixture.

We also experiment with a combination of the phrase pair similarity features de-

[5]The combined topic vectors were renormalised before computing their similarities with each candidate phrase pair.

[6]For a sentence of length 20 the interpolation weights would be set to $\lambda_{local} = 20/30, \lambda_{global} = 10/30$.

rived from the PPT model with a document similarity feature from the pLDA model introduced in Chapter 4. The motivation is that the pLDA model learns topic mixtures for documents and uses phrases instead of words to infer the topical context. Therefore, it might provide additional information to the similarity features described here.

6.6 Data and experimental setup

For ease of comparison, we use the same experimental setup as in Chapter 4 and Chapter 5, with training, development and test data as shown in Table 5.2. The setup allows us to evaluate our dynamic topic adaptation approach because the test documents are from different domains and also differ within each domain, which makes lexical selection a much harder problem. As before, topic adaptation does not make use of the domain labels in training or test, but infers topic mixtures in an unsupervised way. This implies that the model is trained on documents from all three domains without distinguishing between them. However, we compare the performance of our dynamic approach to domain adaptation methods by providing them the domain labels for each document in training and test.

Table 6.3 shows the average length of a document for each domain. While a CC document contains 29.1 sentences on average, documents from NC and TED are on average more than twice as long. The length of a document could have an influence on how reliable global topic information is but also on how important it is to have information from both local and global test contexts.

Data	**CC**	**NC**	**TED**
Test documents	65	31	24
Avg sentences/doc	29.1	60.6	78.9

Table 6.3: Average number of sentences per document in the test set (per domain).

6.6.1 Unadapted baseline system

As before, the baseline is a phrase-based French-English system trained on the concatenation of all parallel data. It was built with the Moses toolkit (Koehn et al., 2007) using the 14 standard core features including a 5-gram language model. Translation quality is evaluated on a large test set, using the average feature weights of three op-

timisation runs with PRO (Hopkins and May, 2011). We use the mteval-v13a.pl script to compute case-insensitive BLEU scores.

6.6.2 Domain-adapted benchmark systems

As domain-aware benchmark systems, we use the linear mixture model (LIN-TM) of Sennrich (2012b) and the phrase table fill-up method (FILLUP) of Bisazza et al. (2011) (both available in the Moses toolkit). For both systems, the domain labels of the documents are used to group documents of the same domain together. We build adapted tables for each domain by treating the remaining documents as out-of-domain data and combining in-domain with out-of-domain tables. For development and test, the domain labels are used to select the respective domain-adapted model for decoding. Both systems have an advantage over our model because of their knowledge of domain boundaries in the data. This allows for much more confident lexical choices than using an unadapted system but is not possible without prior knowledge about each document.

6.6.3 Implementation of similarity features

After all test topic vectors have been computed, a feature generation step precomputes the similarity features for all pairs of test contexts and applicable phrase pairs for translating source phrases in a test instance. The phrase table of the baseline model is filtered for every test instance (a sentence or document, depending on the context setting) and each entry is augmented with features that express its semantic similarity to the test context. We use a wrapper around the Moses decoder to reload the phrase table for each test instance, which enables us to run parameter optimisation (PRO) in the usual way to get one set of tuned weights for all test sentences. It would be conceivable to use topic-specific weights instead of one set of global weights, but this is not the focus of this work.

6.7 Results and discussion

In this section we present experimental results of our model with different context settings and against different baselines. We use bootstrap resampling (Koehn, 2004b) to measure significance on the mixed test set and mark all statistically significant results compared to the respective baselines with asterisk (*: $p \leq 0.01$).

6.7.1 Local context

In Table 6.4[7], we compare the results of the concatenation baseline and a model containing the *phrSim-local* feature in addition to the baseline features, for different numbers of latent topics. We show results for the mixed test set containing documents from all three domains as well as the individual results on the documents from each domain. While all topic settings yield improvements over the baseline, the largest improvement on the mixed test set (+0.48 BLEU) is achieved with 50 topics. Topic adaptation is most effective on the TED portion of the test set where the increase in BLEU is 0.59.

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
10 topics	*27.15	19.87	29.63	32.36
20 topics	*27.19	19.92	**29.76**	32.31
50 topics	***27.34**	**20.13**	29.70	**32.47**
100 topics	*27.26	20.02	29.75	32.40
>Baseline	+0.48	+0.52	+0.34	+0.59

Table 6.4: BLEU scores of baseline system + *phrSim-local* feature for different numbers of topics.

6.7.2 Global context

Table 6.5 shows the results of the baseline plus the *phrSim-global* feature that takes into account the whole document context of a test sentence. While the largest overall improvement on the mixed test set is equal to the improvement of the local feature, there are differences in performance for the individual domains. For Commoncrawl documents, the results vary slightly but the largest improvement is still achieved with 50 topics and is almost the same for both. For News Commentary, the scores with the local feature are consistently higher than the scores with the global feature (0.20 and 0.22 BLEU higher for 20 and 50 topics). For TED, the trend is opposite with the global feature performing better than the local feature for all topics (0.28 and 0.40 BLEU higher for 10 and 20 topics). The best improvement over the baseline for TED is 0.83 BLEU, which is higher than the improvement with the local feature.

[7] which is identical to Table 5.3 in the last chapter

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
10 topics	*27.30	20.01	29.61	32.64
20 topics	***27.34**	20.07	29.56	**32.71**
50 topics	*27.27	**20.12**	29.48	32.55
100 topics	*27.24	19.95	**29.66**	32.52
>Baseline	+0.48	+0.51	+0.24	+0.83

Table 6.5: BLEU scores of baseline system + *phrSim-global* feature for different numbers of topics.

6.7.3 Relation to properties of test documents

To make these results more interpretable, Table 6.6 lists some of the properties of the test documents per domain. Of the three domains, CC has the shortest documents on average and TED the longest. To understand how this affects topic inference, we measure topical drift as the average divergence (cosine distance) of the local topic distributions for each test sentence to the global topic distribution of their surrounding document. There seems to be a correlation between document length and topical drift, with CC documents showing the least topical drift and TED documents showing the most. This makes sense intuitively because the longer a document is, the more likely it is that the content of a given sentence diverges from the overall topical structure of the document.

Property	CC	NC	TED
Per document			
Avg number of sentences	29.1	60.6	78.9
Avg topical divergence	0.35	0.43	0.49
Avg sentence length	26.2	31.5	21.7

Table 6.6: Properties of test documents per domain. Average topical divergence is defined as the average cosine distance of local to global topic distributions in a document.

While this can explain why for CC documents using local or global context results in similar performance, it does not explain the better performance of the local feature for NC documents. However, the last row of Table 6.6 shows that sentences in the NC documents are on average the longest and longer sentences would be expected

to yield more reliable topic estimates than shorter sentences. Thus, we assume that local context yields better performance for NC because on average the sentences are long enough to yield reliable topic estimates. When local context provides reliable information, it may be more informative than global context because it can be more specific.

For TED, we see the largest topical drift per document, which could lead us to believe that the document topic mixtures do not reflect the topical content of the sentences too well. But considering that the sentences are on average shorter than for the other two domains, it is more likely that the local context in TED documents can be unreliable when the sentences are too short. TED documents contain transcribed speech and are probably less dense in terms of information content than News Commentary documents. Therefore, the global context may be more informative for TED which could explain why relying on the global topic mixtures yields better results.

6.7.4 Combinations of local and global context

In Table 6.7 we compare a system that already contains the global feature from a model with 50 topics to the combinations of local and global similarity features described in Section 6.5.

Of the four combinations, the additive combination of topic vectors ($\oplus$) yields the largest improvement over the baseline with 0.63 BLEU on the mixed test set and 0.88 BLEU on TED. The improvements of the combined model are larger than the improvements for each context on its own, with the only exception being the NC portion of the test set where the improvement is not larger than using just the local context.

A possible reason is that when one feature is consistently better for one of the domains (local context for NC), the log-linear combination of both features (tuned on data from all domains) would result in a weaker overall model for that domain. However, if both features encode similar information, as we assume to be the case for CC documents, the presence of both features would reinforce the preference of each and result in equal or better performance. For the additive combination, we expect a similar effect because adding together two topics vectors that have peaks at different topics would make the resulting topic vector less peaked than either of the original vectors.

The additive topic vector combination is slightly better than the log-linear feature

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
+ global	27.27	20.12	29.48	32.55
+ local	*27.43	20.18	29.65	**32.79**
$\oplus$ local	***27.49**	**20.30**	**29.66**	32.76
$\otimes$ local	27.34	20.24	29.61	32.50
$\circledast$ local	*27.45	20.22	29.51	**32.79**
$\oplus$ >BL	+0.63	+0.69	+0.24	+0.88

Table 6.7: BLEU scores of baseline and combinations of phrase pair similarity features with local and global context (significance compared to *Baseline+global*). All models were trained with 50 topics.

combination, though the difference is small. Nevertheless, it shows that combining topic vectors before computing similarity features is a viable alternative to log-linear combination, with the potential to design more expressive combination functions. The multiplicative combination performs slightly worse than the additive combination, which suggests that the information provided by the two contexts is not always in agreement. In some cases, the global context may be more reliable while in other cases the local context may have more accurate topic estimates and a voting approach does not take advantage of complementary information. The combination of topic vectors depending on sentence length ($\circledast$) performs well for CC and TED but less well for NC where we would expect that it helps to prefer the local information. This indicates that the rather adhoc way in which we encoded dependency on the sentence length may need further refinement to make better use of the local context information.

6.7.5 Effect of contexts on translation

To give an intuition of how lexical selection is affected by contextual information, Figure 6.3 shows four test sentences with an ambiguous source word and its translation in bold. The corresponding translations with the baseline, the local and global similarity features and the additive combination are shown in Table 6.8 for the first two examples where the global context yields the correct translation (as indicated by *) and in Table 6.9 for the last two examples where the local context yields the correct

Source	Le **noyau** contient de nombreux pilotes, afin de fonctionner chez la plupart des utilisateurs.
Reference	The precompiled **kernel** includes a lot of drivers, in order to work for most users.
Source	Il est prudent de consulter les pages de manuel ou les faq spécifiques à votre **os**.
Reference	It's best to consult the man pages or faqs for your **os**.
Source	Nous fournissons nano (un petit éditeur), vim (vi amélioré), qemacs (clone de emacs), **elvis**, joe .
Reference	Nano (a lightweight editor), vim (vi improved), qemacs (emacs clone), **elvis** and joe.
Source	Elle a introduit des politiques [..] à coté des **relations** de gouvernement à gouvernement traditionnelles.
Reference	She has introduced policies [..] alongside traditional government-to-government **relations**.

Figure 6.3: Examples of test sentences and reference translations with the ambiguous source words and their translations in bold.

Model	**noyau** $\rightarrow$	**os** $\rightarrow$
Baseline	nucleus	bones
global	kernel*	os*
local	nucleus	bones
global$\oplus$local	kernel*	os*

Table 6.8: Translations of ambiguous source words where global context yields the correct translation (* denotes the correct translation).

Model	**elvis** $\rightarrow$	**relations** $\rightarrow$
Baseline	elvis*	relations*
global	the king	relationship
local	elvis*	relations*
global$\oplus$local	the king	relations*

Table 6.9: Translations of ambiguous source words where local context yields the correct translation (* denotes the correct translation).

translation.[8] In Table 6.8, the additive combination preserves the choice of the global model and yields the correct translations, while in Table 6.9 only the second example is translated correctly by the combined model.

A possible explanation is that the topical signal from the global context is stronger and results in more discriminative similarity values. To verify this hypothesis, we look at two of the examples in more detail. Table 6.10 shows the weighted translation scores for each of the models for the first example from Table 6.8 and Table 6.9, respectively. In example (a), both the global and local model have a preference for the correct translation *kernel*, but the scores of the local model are less discriminative which could be

[8]For these examples, the local model happens to yield the same translations as the baseline model.

noyau →	**kernel***	**nucleus**
global	0.994	0.708
local	0.961	0.867
global⊕local	**0.981**	0.742

(a) Translations of *noyau*

elvis →	**elvis***	**the king**
global	0.711	0.872
local	0.930	0.916
global⊕local	0.815	**0.861**

(b) Translations of *elvis*

Table 6.10: Weighted translation scores for first example from Table 6.8 and Table 6.9, respectively (* denotes the correct translation).

why it has picked the wrong translation. In the combined model, the scores are more discriminative than in the local model and the correct translation appears in the output. In example (b), the global model has a preference for the wrong translation *the king* and its scores are more discriminative than those of the local model which has a preference for the correct translation *elvis*. Thus, the combined model is more influenced by the global model and maintains the preference for the wrong translation. A useful extension could be to try to detect for a given test instance which context provides more reliable information (beyond encoding sentence length) and boost the topic distribution from that context in the combination.

6.7.6 Comparison with domain adaptation

Table 6.11 compares the additive model (⊕) to the two domain-adapted systems that know the domain label of each document during training and test. Our topic-adapted model yields overall competitive performance with improvements of 0.37 and 0.25 BLEU on the mixed test set, respectively. METEOR scores for these experiments can be found in Appendix C, Table C.1. While it yields slightly lower performance on the NC documents, it achieves equal performance on TED documents and improves by up to 0.94 BLEU on Commoncrawl documents. This can be explained by the fact that Commoncrawl is the most diverse of the three domains with documents crawled from all over web, thus we expect topic adaptation to be most effective in comparison to domain adaptation in this scenario. Our dynamic approach allows us to adapt the similarity features to each test sentence and test document individually and is therefore more flexible than cross-domain adaptation approaches while requiring no information about the domain of a test instance.

Type of adaptation	Model	Mixed	CC	NC	TED
Domain-adapted	LIN-TM	27.24	19.61	29.87	32.73
	FILLUP	27.12	19.36	29.78	32.71
Topic-adapted	global⊕local	***27.49**	**20.30**	29.66	**32.76**
	>LIN-TM	+0.25	+0.69	-0.21	+0.03
	>FILLUP	+0.37	+0.94	-0.12	+0.05

Table 6.11: BLEU scores of translation model using similarity features derived from PPT model (50 topics) in comparison with two (supervised) domain-adapted systems.

6.7.6.1 WADE evaluation

Similar to Section 4.6.8, we want to take a closer look at the results on the TED test set using the WADE framework. LIN-TM and the topic-adapted model are scored very similarly on the TED test set by BLEU and METEOR, thus we want to investigate whether there is a qualitative difference between the systems on different subsets of source words.

Table 6.12 compares the relative improvement in terms of correctly translated words of LIN-TM (the stronger domain-adapted system) and the topic-adapted *global⊕local* system. The improvements of the topic-adapted system are larger on all four word sub-classes and in particular they are slightly larger for content and high entropy words in comparison to function and low entropy words. Thus, while yielding overall similar scores, there seems to be a subtle difference between the systems in terms of their effect on different types of source words. Though the results need further validation, they indicate that the topic-adapted system may be better at translating ambiguous words than the domain-adapted system.

	Baseline	**+LIN-TM**	**+ global⊕local**
	% correct	% improvement	
Content words	50.28	0.31	**0.54**
Function words	63.51	**0.39**	0.53
High entropy words	44.81	0.34	**0.66**
Low entropy words	68.69	**0.41**	0.55

Table 6.12: Percentage of correctly translated words with the baseline system and improvements of domain-adapted and topic-adapted models over the baseline according to WADE. Source words are grouped by different sub-classes.

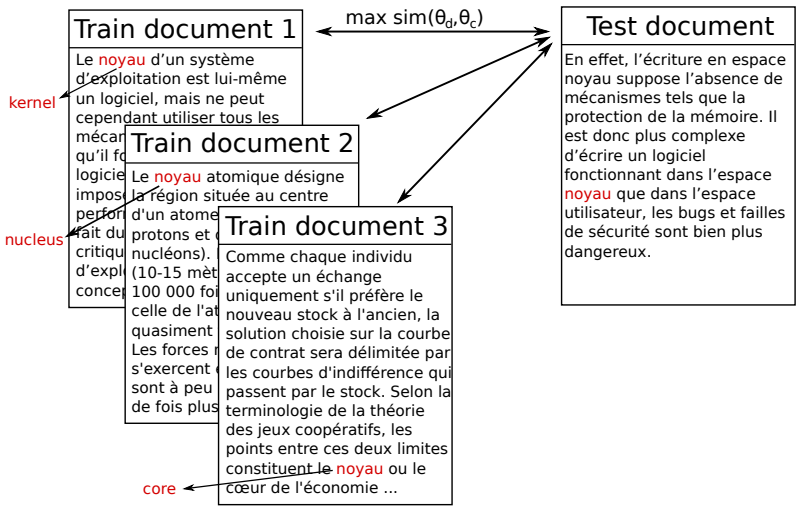


Figure 6.4: The *docSim* feature computes a *maximum similarity* score for each applicable phrase pair, e.g. *noyau* → *kernel*, *noyau* → *nucleus*, *noyau* → *core*.

6.7.7 Combination with an additional document similarity feature

To find out whether similarity features derived from different types of topic models can provide complementary information, we add the *phrSim* features to a system that already includes a document similarity feature (*docSim*) derived from the pLDA model which learns topic distributions at the document level and uses phrases instead of words as the minimal units. This feature computes the maximum topical similarity of the test document to one of the training documents where an applicable source phrase was found, as visualised in Figure 6.4.

The results are shown in Table 6.13. Adding the two best combinations of local and global context from Table 6.7 yields the best results on TED documents with an increase of 0.63 BLEU over the *Baseline+docSim* model and 1.15 BLEU over the baseline. On the mixed test set, the improvement is 0.38 BLEU over the *Baseline+docSim* model and 0.74 BLEU over the baseline. Again, the corresponding METEOR scores can be found in Appendix C, Table C.2. Thus, we show that combining different scopes and granularities of similarity features consistently improves translation results and yields larger gains than using each of the similarity features alone.

6.7.8 Potential improvements

There are a few aspects of our current model that would be worth exploring further in the future. One concerns the representation of the distributional profiles that are used to learn the topical representations. Because very frequent phrases tend to have

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
+ docSim	27.22	20.11	29.63	32.40
+ phrSim-global $\oplus$ phrSim-local	*27.58	20.34	**29.71**	32.96
+ phrSim-global $\circledast$ phrSim-local	***27.60**	**20.35**	29.70	**33.03**
global$\circledast$local>BL	+0.74	+0.74	+0.38	+1.15

Table 6.13: BLEU scores of baseline, baseline + document similarity feature and additional phrase pair similarity features (significance compared to *Baseline+docSim*). All models were trained with 50 topics.

large distributional profiles, topic inference can be quite slow when running on the distributional profiles of all possible phrase pairs. However, since very frequent phrase pairs are not expected to be particularly sensitive to topic changes, a simple heuristic could be to just set their topic mixtures to a uniform distribution and exclude them from the inference. Alternatively, a corpus sampling method could be used to limit the number of training examples used for frequent phrases, but this would equally force us to define frequency thresholds.

6.7.9 Relation to Findings on Word Sense Disambiguation

In this section, we compare our findings about contextual information for MT to the findings in the WSD literature where the difficulties underlying the task are very similar to those we address for MT. Much of the work in the WSD literature relies on feature sets similar to the following (Agirre et al., 2005a):

Local Collocations Bigrams and trigrams around the target word (surface forms, lemmas, POS tags), bigrams and trigrams with the previous or following lemma/surface word, content words in a short window around the target word.

Syntactic Features Dependencies such as object, subject, noun-modifier, preposition, sibling lemmas.

Bag-of-words Features Lemmas of content words in the entire context, salient bigrams in the context.

Agirre et al. (2005a) explore feature set combinations for WSD and compare the performance of features and feature groups. They observe that across different classifiers the local collocation features generally performed best and slightly better than the bag-of-words features[9]. Both feature sets in turn performed better than the syntactic and bag-of-bigrams features.

Gliozzo et al. (2005) employ kernel methods for WSD where each kernel provides a measure of similarity of the features associated with a word sense and the features found in the context of a word occurrence. They demonstrate the benefit of a *domain kernel* that captures the topic of a given context using LSA. The domain kernel performs well on its own, though slightly weaker than the word collocation kernel[10]. Comparing a set of all features to a set of all features except the domain kernel shows that adding the domain kernel improves performance substantially over the remaining features. This demonstrates that domain information is very informative for sense disambiguation, even when the classifier already contains features from the same context (BOW features). Agirre et al. (2005b) and Agirre and Lopez De Lacalle (2008) provide further evidence in favour of dimensionality reduction techniques to alleviate feature sparsity.

Cai et al. (2007) show that topic features from the context of a target word improve performance over the standard WSD feature set in a Bayesian Network, for the *lexical sample* and *all words* tasks. A further experiment that compares the performance of support vector machines with 1) the standard features, 2) the (standard + topic) features and 3) the (standard + topic - bag-of-words) features shows a 1.2 improvement in accuracy when adding in the topic features[10]. Taking out the bag-of-words features results only in a 0.1 decrease in accuracy which shows that the topic features are more informative than the bag-of-words features.

Li et al. (2010) use topic models to compare test contexts and WordNet sense paraphrases in order to select the correct WordNet sense. Their system solely relies on topic similarity and does not use the standard WSD features. They provide an interesting comparison of experiments with different context sizes[11] which can be summarised with the following list ordered by F1 performance which shows that sentence context for topic inference performs better than smaller contexts within the same sentence but

[9]Experiments were run on the Senseval-2 English lexical sample task where the context consists a a small paraphgraph of text preceding the sentence containing the target word.

[10]Experiments were run on the Senseval-3 English lexical sample task where the context consists of small paragraph of 1-2 sentence before and after the sentence containing the target word.

[11]Experiments were run on the Semeval-2007 English coarse-grained all words task where the context consists of an entire article.

expanding the context to the entire text causes a hit in performance:

$$\text{sentence context} > \pm 10 \text{ words} > \pm 5 \text{ words, entire text} > \pm 1 \text{ word}$$

The maximum available context for this experiment was larger than the contexts used for the previously mentioned works which were run on lexical sample tasks with paragraph context. Thus, it may be that the content of the articles in the Semeval-2007 all words task was too dispersed, especially given that the number of topics used in this work is rather high (from 150 to 1000 topics). It is likely that the smaller numbers of topics used in our experiments (from 10 to 100 topics) are more robust to noisy contexts.

In summary, these findings suggest that

- local collocations are very strong features for WSD
- additional bag-of-words features improve WSD performance
- reducing the dimensionality of bag-of-words features further improves performance
- context sizes larger than the sentence or paraphraph context may decrease performance of topic model approaches, but evidence for this is limited

In the light of these findings, our choice of using topic information to improve sense disambiguation for SMT seems reasonable. The baseline SMT system already contains local collocational information in the source and target phrase contexts as well as in the language model contexts. What is usually missing in standard MT systems are features from the wider source context which correspond to the bag-of-words or topic features of the WSD literature. Note that our usage of the words *local* and *global* differs from the usage in the WSD literature. Both our local and global features correspond to the global (bag-of-words) features in the WSD literature. However, we distinguish between two scopes of this context, the local, sentence-level scope and the global, document-level scope which can contain diverging information. Though work by Li et al. (2010) suggests that sentence-level context is best for topic inference, we believe that the nature of the documents, the number of topics in the model as well as the task can have a strong influence on the optimal context size. Another possible reason for the fact that we see good MT performance using document-level context for topic inference is that our task is more coarse-grained than typical WSD tasks. Even the

coarse-grained lexical sample task used for Li et al.'s context comparison uses sense definitions that are likely to be more fine-grained than those in a typical MT task where multiple dictionary-style word senses can share the same translation.

In the SMT literature, Carpuat (2009) argues in favour of a document-level consistency constraint for translations within the same document. Mei (2010) have provided evidence for the usefulness of *global*, document-level WSD using a graph-based method that takes into account all content words in a document. Our work in this chapter and the previous chapters supports the hypothesis that abstract, topical representations are useful for disambiguation and improve performance over existing contextual information in standard SMT systems. We also show that both sentence-level and document-level context can be suitable for capturing the semantics underlying translation choices.

6.8 Conclusion

In this chapter, we have reviewed past approaches to taking into account local and global contextual information inside and outside the field of statistical machine translation. We have presented an extension of the PPT model from Chapter 5 to integrate both levels of context for dynamic model adaptation at test time.

Our experimental results show that it is beneficial for adaptation to use contextual information from both local and global contexts, with BLEU improvements of up to 1.15 over the baseline system on TED documents and 0.74 on a large mixed test set with documents from three domains. Among four different combinations of local and global information, we found that the additive combination of topic vectors performs best. We conclude that information from both contexts should be combined to correct potential topic detection errors in either of the two contexts. We also show that our dynamic adaptation approach performs competitively in comparison with two supervised domain-adapted systems and that the largest improvement over these systems is achieved for the most diverse portion of the test set.

We have linked our results to findings from the WSD literature that has explored the relationship between local collocational features, bag-of-words features and representations thereof in lower-dimensional space. It was established that bag-of-words features improve performance on top of collocational features and that abstracting from the lexical forms of these features further increases performance. Therefore, we argue that adding topic features on top of standard SMT systems that already make use of

local collocational information is a natural extension and in line with the findings of the WSD literature. We have further shown that it can be beneficial to take into account different scopes of topical context which can provide complementary information.

Chapter 7

Combining Multi-domain Adaption with Topic Adaptation

In Chapters 4-6 we followed the assumption that no domain information is available to the translation system, neither at training nor at test time. However, this assumption could be relaxed because even though we cannot assume domain labels at test time in our scenario, we usually have some form of information about our training data. For example, if we have used the TED corpus, we know which parts of our training data contain transcribed speeches.

One open question related to domain adaptation concerns the relationship between text *style* (e.g. formal vs. informal) and text *genre* (e.g. news article vs. legal text). A widespread definition of *domain* is that it denotes the source of a text corpus, such as Europarl or TED. Style and genre characterise a text along different axes and often the difference between two domains is characterised by changes in both style and genre. If a translation model is trained on News Commentary text and tuned on Newswire text, one could argue that it is only adapted along the *genre* axis. On the other hand, differences in style can be the result of genre change and thus *style* may be the more decisive factor that influences translation changes. On yet another axis there are thematic or topical changes which - as we have argued in previous chapters - can be independent of corpus boundaries. One hypothesis is therefore that when a corpus varies along both the stylistic and the thematic axis, domain adaptation captures the stylistic commonalities while topic adaptation captures the thematic differences and therefore differences in meaning.

So far, the topic adaptation approaches described in this thesis as well as in the literature have typically not made use of domain information and it is not clear whether

domain information can be a useful addition once we have learned unsupervised topics. While topic models are the method of choice for detecting and grouping the semantic differences in documents, making use of our knowledge about the different corpora in the training data could potentially help to adapt more specifially to *style* on top of adapting to topics. By predicting the domain label of test documents, we can combine both approaches to translate unlabelled documents of different styles and topics.

Therefore, in this chapter we explore the following questions:

1. Can the combination of domain adaptation and topic adaptation be beneficial?
2. Is there evidence that domain and topic adaptation differ in capturing stylistic and topical variation in documents?
3. Can the representations learned by topic models be used to automatically predict the domain of a document?

7.1 Related work

An extension of the standard domain adaptation task is *multi-domain adaptation* where a translation system is adapted to several known target domains (see for example Cui et al. (2013)). In cases where the target domains are not assumed to be known, dedicated domain classifiers can be trained and used to automatically predict the target domain and choose an appropriate model (Banerjee et al., 2010).

Domain classification for multi-domain adaptation has been the focus of several researchers in recent years. Xu et al. (2007) tune domain-specific feature weights and build domain-specific language models. They use the perplexity of in-domain language models to classify test documents and select the appropriate weights and models per document. Banerjee et al. (2010) train domain-specific translation models and use SVMs to detect the domain of an input sentence to route it to a domain-specific model. Wang et al. (2012) follow a slightly different approach by re-using the same translation model for all domains and tuning domain-specific features weights with modified objectives. Sennrich et al. (2013) adapt the four standard translation model features to unsupervised clusters of the development data obtained by k-means clustering. Their method can be seen as noise-robust multi-domain adaptation with known domains since the adaptation development set contains in-domain data from all test domains which are recovered by the clustering algorithm.

Another line of research aims to improve topic modelling by encoding domain information via a Dirichlet Forest Prior (Andrzejewski et al., 2009). By specifying Must-Link and Cannot-Link relations between words, topic modelling is guided to either separate words into different topics or merge them into the same topic. While the idea of combining domain and topic adaptation within the same model is appealing, the model requires manually constructed lists of words and seems more suited for fine-tuning specific topics, a process they call *interactive topic modeling*.

Different from previous work, we propose to combine multi-domain adaptation with topic adaptation to adapt to both style and topic in a test document of unknown origin, without using domain-specific development data for adaptation[1]. We also show that topic modelling makes it straightfoward to predict the domain of a test document, circumventing the need for separately trained domain classifiers. This allows us to combine domain-adapted translation models with topic-adapted models dynamically at test time.

7.2 Topic modelling approach

We follow the approach described in Chapter 5 to build a phrase pair topic model (PPT) with 50 topics[2]. The model learns topic vectors for all phrase pairs from distributional profiles[3] and compares them to the topic representation of a test context, in our case the document context, by measuring their cosine similarity. Each phrase pair in the phrase table receives additional features depending on its topical similarity with the test document and thus the topic-adapted features are specific to each test document.

7.2.1 Topic features

Our work in Chapter 4 showed that combining several adapted features can improve performance over only a single adapted feature. It also showed that the probabilistic adapted feature resulted in the best performance compared to three other types of adapted features. Therefore, we want to expand the PPT model by adapting other features on top of the similarity feature. In particular, given the results in Chapter 4 we

[1] While we do adapt the domain-specific models using in-domain development sets,the approach does not rely on the test documents matching the same domains as the domain labels are predicted automatically.

[2] Other work in the literature as well as our own previous work has shown that 50 latent dimensions are often reasonable for semantic representations.

[3] Pseudo-documents built from the context words of a phrase pair, see Section 5.3.

want to attempt to derive probabilistic translation features from this model. In the following, we consider several sets of topic features containing the individual features described below, where s and t denote a source and target phrase, c denotes the test document context, k denotes a latent topic and θ denotes a topic vector:

Conditional translation probability

$$\begin{aligned} P(t|s,c) &= \sum_k P(t,k|s,c) \\ P(t,k|s,c) &\propto P(t,s,k|c) \\ &= P(t|s,k) \cdot P(s|k) \cdot P(k|c) \end{aligned} \tag{7.1}$$

Joint-conditional probability

$$\begin{aligned} P(t,c|s) &= P(c|t,s) \cdot P(t|s) \\ &\approx P(\theta_c|\theta_{pp}) \cdot P(t|s) \\ &\approx cos(\theta_c|\theta_{pp}) \cdot P(t|s) \end{aligned} \tag{7.2}$$

Target-unigrams

$$trgUnigrams_t = \prod_{i=1}^{|t|} f(\frac{P_{doc}(w_i)}{P_{baseline}(w_i)}) \cdot f(\frac{P_{doc}(w_i)}{P_{topic0}(w_i)}) \tag{7.3}$$

Sim-phrasePair

$$similarity = cosine(\theta_{pp}, \theta_c) \tag{7.4}$$

Sim-targetPhrase

$$similarity = cosine(\theta_{tp}, \theta_c) \tag{7.5}$$

Sim-targetWord

$$similarity = cosine(\theta_{tw}, \theta_c) \tag{7.6}$$

The first two features are probabilistic features that take the topical context into account in computing the probability of a target phrase given a source phrase. The first feature, **Conditional**, factorises the joint probability of a target phrase t, source phrase s and topic k given a context c into the probabilities $P(t|s,k)$, $P(s|k)$ and $P(k|c)$, similar to the formulation in Section 4.3. The first two probabilities are estimated from relative counts of how often source and target phrases co-occur with each topic

in the distributional profiles[4], and $P(k|c)$ represents the inferred topic mixture for the test context. The second feature, **Joint-conditional**, estimates the joint probability of a target phrase and a test context given a source phrase. It is factorised as the (baseline) probability of a target phrase given a source phrase and the probability of the test context given the source and target phrase. The latter is approximated by the probability of the test context topic mixture given the phrase pair topic mixture, which is further approximated by the cosine similarity between the two topic mixtures.

The **Target-unigrams** feature is inspired by the lazy MDI adaptation of Ruiz and Federico (2012) and measures the probability ratio of a word under the document topic mixture versus under the baseline model[5]. As in Chapter 4, we include an additional term to measure the topical relevance of a word by comparing against its probability under the asymmetric topic 0 of the PPT model[6]. **Sim-phrasePair** measures the cosine similarity of a phrase pair topic vector and the topic vector of a test context, as defined in Chapter 5. **Sim-targetPhrase** is similar but uses an average topic vector over all phrase pairs that match the target phrase. **Sim-targetWord** instead replaces the phrase pair topic vector with the word topic vector of the word in the target phrase with the lowest topical entropy[7]. Target word topic vectors are derived from phrase pair topic vectors by averaging over all vectors of phrase pairs that include the target word.

An important difference between probabilistic and similarity features in the PPT model is that while the probabilistic features have some notion of the frequency of translations in the training corpus (which is implicit in the number of context words in a distributional profile), similarity features are purely based on topic information and could be unreliable for rare or singleton translation units.

For the adaptation experiments, we evaluate a topic feature set that contains all the features above except the combined similarity feature, as well as smaller subsets of the features. The large feature set overlaps with the unadapted and domain-adapted features sets in that each contains probabilistic translation features. The smaller sets do not overlap with the baseline feature sets in that respect because they only contain features that have no correspondence in the baseline models. We argue that this differ-

[4]Note that the counts used for computing these probabilities are the document-topic counts for pseudo-documents collected in training. Pseudo-documents correspond to pairs of source and target phrases.

[5]The baseline model here corresponds to the relative frequency of target unigrams in the training data.

[6]Topic 0 has higher a priori probability and is supposed to capture common words that occur in the context of many translation units.

[7]The intuition behind this feature is that words with low topical entropy are expected to be more topically relevant.

ence has an influence on the relative improvement of combining domain-adapted and topic-adapted features. The reason is that when features encode similar information for many development examples, the tuning procedure may assign a high feature weight to one of the features while the other features receive small weights. The topic feature sets are defined as:

Overlap Conditional, Joint-conditional, Target-unigrams, Sim-phrasePair, Sim-targetPhrase, Sim-targetWord
Sim-combine similarity = $\frac{1}{3}$ (sim-pp + sim-tp + sim-tw)
Sim-combine-loglin Sim-phrasePair, Sim-targetPhrase, Sim-targetWord
Sim-combine+trgUnigrams Sim-combine, Target-unigrams

7.3 Predicting domain labels

While previous approaches to automatic domain classification have built dedicated classifiers such as SVMs and perceptrons or used in-domain language model perplexity, we re-use our already trained topic models to assign domain labels to documents[8]. We apply the phrase pair topic model to all documents from the three training domains (CC, NC, TED) to get one topic vector per training document. We then experiment with three types of nearest-neighbour classifiers using the induced topic vectors:

Single-prototype Compute the average of all document vectors of the same training domain ($\rightarrow$ domain vectors), then compute the cosine similarity of a test document with the three domain vectors and predict the domain with the highest similarity.

Multi-prototype Compute the cosine similarity of a test document with the topic vectors of all training documents and predict the domain according to the label of the most similar training document.

Single-prototype-threshold Like single-prototype but with a threshold of 0.5 for prediction[9]. If a test document is not similar to any of the domain vectors according to the threshold, predict "unknown" and use the baseline model in place of a domain-

[8]Blei et al. (2003) propose the use of LDA models for document classification. However, they use the document-topic distributions of training documents as features in an SVM. They show empirically that using topic features yields overall better classification accuracy than using word features.

[9]Cosine similarity ranges from 0 to 1.

Model	**CC**		**NC**		**TED**	
# dev+test docs	88		39		24	
	sgl	mlt	sgl	mlt	sgl	mlt
k=10	0.70	**0.88**	**1.0**	0.95	**1.0**	0.96
k=20	0.82	**0.94**	**1.0**	0.97	**1.0**	1.0
k=50	0.73	**0.93**	**1.0**	0.95	**1.0**	1.0
k=100	0.76	**0.93**	**1.0**	1.0	**1.0**	0.92

Table 7.1: Accuracy of domain prediction using single-prototype (sgl) or multi-prototype (mlt) domain vectors with different numbers of topics (k).

Model	**CC**			**NC**			**TED**		
# dev+test docs	88			39			24		
	corr	other	unk	corr	other	unk	corr	other	unk
k=10	0.64	0.27	0.09	1.0	0.0	0.0	1.0	0.0	0.0
k=20	0.50	0.09	0.41	1.0	0.0	0.0	1.0	0.0	0.0
k=50	0.34	0.05	0.61	0.93	0.0	0.07	0.96	0.0	0.04
k=100	0.26	0.05	0.69	0.87	0.0	0.13	0.92	0.0	0.08

Table 7.2: Accuracy of domain prediction using single-prototype vectors with a threshold of 0.5 and different numbers of topics (k). *Corr*: correct domain predicted, *other*: wrong domain predicted, *unk*: no domain predicted.

adapted model.

The results of the single- and multi-prototype classifiers on the development and test documents are shown in Table 7.1. While for NC and TED documents, we can get perfect domain predictions with the single-protoype classifier, the accuracy on CC is at most 0.82, depending on the number of latent topics in the topic vectors. However, the multi-prototype classifier does better for CC in all cases. This suggests that there are subclusters of documents in the CC corpus to which some of the CC test documents are similar while not being as similar to a global average of all CC documents. Table 7.2 shows the accuracy of the single-protoype classifier when using a fixed threshold, with the results split into *correct*, *other* and *unknown*. While NC and TED documents are still labelled accurately (particularly for k=10 and k=20), the proportion of correct predictions drops for CC. This leads to the conclusion that NC and

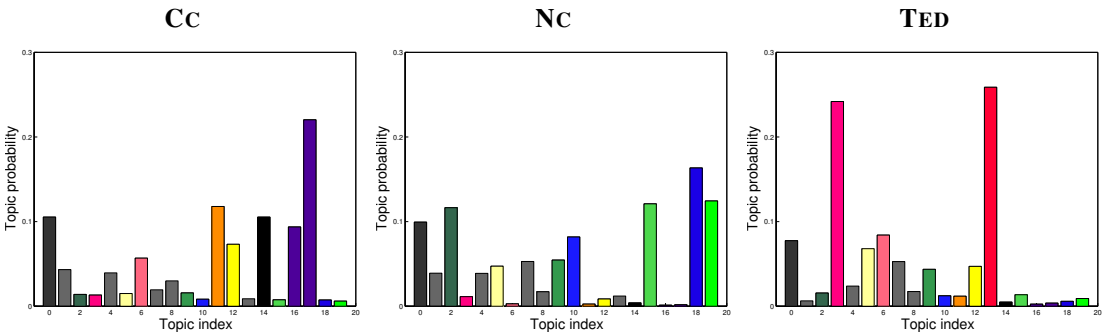


Figure 7.1: Average domain vectors (20 topics) for Commoncrawl, News Commentary and TED corpus. Correspondence of topic indices to topic labels: politics=2,9,15, speech=3,6, health=5, climate=10, IT=11, arts=12, science=13, English text=14, hotels=16,17, economy=18, war=19.

TED can be considered real domains in the sense that the documents all have certain properties in common, while this does not seem to be the case for CC. This is also supported by Figure 7.1 which shows the average domain vectors for each of the three corpora and provides labels for some of the topical peaks according to their most likely words. While CC documents can belong to rather diverse clusters such as IT, arts, hotel reviews or speech, NC documents belong to more related topics along the themes of politics and economy. These topics are more likely to be active within the same document and thus a document with political or economical content would likely overlap with the NC domain vector on several dimensions. TED documents share two topical components that capture words that are typical in speech like 1st and 2nd person verb forms (*speech*) as well as a rather broad *science* topic. Thus, a document with a high proportion of these verb forms would be likely to be classified as TED.

Another interesting observation from Table 7.2 is that the prediction accuracy seems to be inversely correlated with the number of latent topics in the domain vectors and document topic vectors. The reason for this effect is that we use a fixed, untuned classification threshold while the cosine similarities of higher-dimensional topic vectors are typically lower than those of low-dimensional vectors[10]. In fact, if we lower the threshold to 0.35, we can regain perfect classification accuracy for NC and TED for all reported values of k. However, we prefer not to tune the threshold and simply select the classifier that yields the best predictions on the development set, which is the classifier with k=20.

[10]This trend was observed by Banchs and Costa-jussà (2011) for vectors derived from Latent Semantic Indexing.

7.4 Experimental setup

We use the same training, development and test data as in previous chapters, which is described in Section 4.5.1. All of the corpora contain document boundaries which allows us to consider document context during translation and switch translation and language models at document boundaries. While the domain-adapted baselines use gold domain labels, we use automatically predicted domains when combining domain-adapted and topic-adapted models[11]. We use a tuning set containing data from all three domains and use one set of tuned feature weights for all portions of the test set. Translation quality is evaluated using the average feature weights of three optimisation runs with PRO (Hopkins and May, 2011). We use the mteval-v13a.pl script to compute case-insensitive BLEU scores.

7.4.1 Unadapted baseline system

Our baseline is a phrase-based French-English system trained on the concatenation of all parallel data. It was built with the Moses toolkit (Koehn et al., 2007) using the 14 standard core features including a 5-gram language model, trained on the concatenation of the target training data.

7.4.2 Domain-adapted systems

We use the linear mixture model (DA-TM) of Sennrich (2012b) (available in the Moses toolkit) to adapt the translation model to each of the three domains CC, NC and TED. The domain labels of the documents are used to group documents of the same domain together. We build adapted tables for each domain by treating the remaining documents as out-of-domain data. For development and test, the domain labels are used to select the respective adapted model for decoding. We also use domain-adapted language models (DA-LM) which are linear interpolations of separate language models, tuned to minimise perplexity on an in-domain development set per domain.

7.4.3 Topic-adapted systems

In order to integrate document-specific features into decoding, we build a (filtered) phrase table with topic-adapted features for each test document which is loaded before

[11] Note that topic adaptation does not rely on domain labels.

decoding each document. It would be straightforward to achieve a tighter integration with the SMT system by setting up feature functions that have access to document-level information, but for now we use a simple architecture where a wrapper script runs the decoder for each document.

7.4.4 Systems combining domain and topic adaptation

We use a simple approach to combine domain and topic adaptation by log-linear combination of features. When combining with the domain-adapted translation model, the topic-adapted features are added to the already domain-adapted phrase table. Combining with the domain-adapted language model is done by switching the baseline language model for the domain-adapted language model.

7.5 Results

In this section we present results of different combinations of the baseline model, the domain-adapted and the topic-adapted models. Results are reported separately per test domain as well as on the entire mixed test set.

7.5.1 Overlapping topic feature set

Table 7.3 shows the results when adding the overlapping topic feature set (containing probabilistic and non-probabilistic translation features) on top of unadapted and domain-adapted systems. Adding topic-adapted features always yields improvements over the respective baseline system, even though the level of previous adaptation has an influence on the relative gain. We use bootstrap resampling (Koehn, 2004b) to measure significance of the BLEU scores on the mixed test set and mark all statistically significant results compared to the respective baseline system with asterisk (*: $p \leq 0.001$).

Topic adaptation works best for TED documents but we observe that the improvement decreases with increasing domain-adaptation. Depending on the level of previous adaptation, the BLEU improvements range between 1.34 and 0.31. These results add to our observations from Section 7.3 that on top of acting as a domain, TED documents exhibit a further layer of structure that can be exploited with topic adaptation. For CC, the improvement of topic adaptation is quite stable at between 0.6 and 0.7 BLEU because domain adaptation has almost no effect on performance here. This is

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
+ topics	*27.57	20.35	29.68	33.22
	+0.71	+0.74	+0.26	+1.34
DA-TM	27.24	19.61	29.87	32.73
+ topics	***27.73**	**20.33**	**29.88**	**33.55**
	+0.49	+0.69	+0.01	+0.82
DA-LM	27.16	19.71	29.77	32.46
+ topics	*27.60	20.37	29.80	33.20
	+0.44	+0.63	+0.03	+0.74
DA-TM+LM	27.34	19.59	29.92	33.02
+ topics	*27.63	20.22	29.90	33.33
	+0.29	+0.60	-0.02	+0.31
Gain of best system over baseline	+0.87	+0.72	+0.46	+1.67

Table 7.3: BLEU results of unadapted/adapted baseline models and additional topic-adapted features (Overlap) with their gain over the respective baseline (bottom of each box). The best system on the mixed test set is marked in bold. *: $p \leq 0.001$ marks significantly better scores compared to the respective baseline.

in line with our observations from Section 7.3 that CC behaves least like a domain in comparison with the other two corpora. For NC documents, the topic-adapted features yield a small improvement of 0.24 BLEU over the unadapted system but no further improvement over the domain-adapted models. A possible explanation is that because of the close relation between the dominant topics in the NC corpus (politics/economy), domain adaptation methods are sufficient to capture the important characteristics of the documents. Overall, the best results on the mixed test set are achieved with a combination of domain and topic adaptation of the translation model (DA-TM + topics). On the mixed test set, the gain of this model over the DA-TM model is 0.49 and the gain over the unadapted baseline system is 0.87 BLEU. On TED documents, it yields a 0.82 BLEU improvement over the DA-TM model and a 1.67 improvement over the unadapted baseline. METEOR scores for these experiments can be found in Appendix D, Table 7.3. Even though the absolute improvements are smaller than for BLEU, the overall trend is similar and the same system is ranked best.

An indicator that the probabilistic features in the domain-adapted and topic-adapted

Model	Domain-adapted features					Topic-adapted features					
DA-TM + topics	0.030	0.042	-0.021	0.023	0.098	0.040	0.053	0.111	-0.002	-0.004	0.045
- probabilistic	0.038	0.042	0.077	0.027	0.126	-	-	0.093	0.042	-0.024	0.042

Table 7.4: Tuned feature weights of combined domain-adapted and topic-adapted features, with and without probabilistic topic-adapted features (Conditional/Joint-Conditional). All probabilistic forward translation features are marked in red.

models do overlap are the tuned feature weights for these models, as shown in Table 7.4. In the first line, both **Conditional** and **Joint-Conditional** features are active and the domain-adapted feature $P(e|f)$, which is usually an important translation features, receives a negative weight. In contrast, when these features are not active in the model, the domain-adapted feature $P(e|f)$ receives a large positive weight, which indicates that its weights depends on whether other features have a similar function.

Experimental results for the **Conditional** and **Joint-Conditional** features in isolation can be found in Appendix D, Table D.1.

7.5.2 Smaller topic feature sets

While the results from the previous section show that topic adaptation is beneficial at all levels of domain adaptation as long as the test documents are "topic-adaptable" (CC and TED), the role of domain adaptation is not that clear yet as the difference between the best topic-adapted system with and without domain-adapted features is relatively small (27.73 vs. 27.57 BLEU on the mixed test set and 33.55 vs. 33.22 on TED). Therefore, we study the effect of adding domain-adapted features to already topic-adapted systems with smaller topic feature sets, thereby avoiding overlap between the feature sets. In this setup, we would expect larger gains from adding the domain-adapted features than in Table 7.3. Another goal is to measure the contribution of particular topic features and to find the most efficient combination of topic and domain features, both in terms of qualitative performance and in terms of computational effort, which can be an issue in dynamic adaptation[12].

Table 7.5 shows the results when adding the domain-adapted features to the topic feature sets that only contain non-probabilistic features that do not overlap with the domain-adapted features. The upper part of the table shows the performance with sin-

[12]Computing similarity features is much faster than computing other topic-adapted features, partly because they do not require computing normalisation constants.

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
+ TrgUnigrams	27.04	19.86	29.25	32.57
+ DA-TM	**27.50	19.96	29.77	33.34
+ Sim-phrasePair	27.32	20.19	29.31	32.66
+ DA-TM	27.53	20.04	30.05	32.98
+ Sim-targetPhrase	27.21	19.92	29.39	32.58
+ DA-TM	**27.52	19.96	29.94	33.20
+ Sim-targetWord	26.99	19.89	29.16	32.12
+ DA-TM	**27.44	19.91	29.98	32.94
+ Sim-combine	27.29	20.10	29.49	32.60
+ DA-TM	**27.69	20.13	29.90	33.37
+ Sim-combine-loglin	27.18	20.13	29.55	32.34
+ DA-TM	*27.41	19.93	29.86	32.97
+ Sim-combine+trgUnigrams	27.21	20.05	29.36	32.78
+ DA-TM	**27.47	19.87	29.76	33.36
DA gain of best system	+0.40	+0.03	+0.41	+0.77
Gain of best system over baseline	+0.83	+0.52	+0.48	+1.49

Table 7.5: BLEU results of smaller topic feature sets with added domain-adapted features. The best system on the mixed test set is marked in bold and its improvements over the topic-adapted system and the baseline are shown at the bottom of the table. **: $p \leq 0.01$, *: $p \leq 0.05$ mark significant improvements over a topic-adapted system.

gle topic features, the lower part shows combinations of two or three topic features. In all experiments, the topic features improve over the unadapted baseline and the additional domain-adapted features improve over the topic-adapted model on the mixed test set. Among the single topic features, the **Sim-phrasePair**[13] feature yields the best performance on the mixed test set (27.32) and this trend persists when adding the domain-adapted features (27.53). Note that the performance of the features varies with respect to the domain of the test set, for example *Baseline+Sim-phrasePair+DA-TM* yields

[13]Comparing with the results in Appendix D, Table D.1 shows that the **Sim-phrasePair** feature and the **Conditional** feature in isolation yield very similar performance (27.32 vs. 27.34 BLEU on the mixed test set). This shows that while encoding the topic information in quite different ways in these two features, the effect on translation quality is similar.

the best performance on NC documents (30.05) while *Baseline+TrgUnigrams+DA-TM* yields the best performance on TED documents (33.34).

The best overall performance is achieved with the **Sim-combine** feature. It yields similar performance to **Sim-phrasePair** on its own and better performance when combined with the domain-adapted features. For this setup, the added gain of domain adaptation on top of topic adaptation is 0.77 BLEU on TED, 0.41 on NC and 0.40 overall (as shown in the second last row of the table). The overall performance of this combined model with a smaller topic feature set is almost as good as the best model from Table 7.3, with a 0.83 BLEU improvement on the mixed test set and a 1.49 improvement on TED documents over the unadapted baseline. METEOR scores for these experiments show a similar relation between the different setups and can be found in Appendix D, Table D.3.

These results shows that combining topic adaptation with domain adaptation is particularly useful when each model deals with different aspects of translation, as is the case for probabilistic features and similarity features. As topic adaptation requires dynamic computation at test time, an architecture where part of the adaptation is done offline can reduce computational effort at test time. This can be done by computing only a small set of topic features and using domain-adapted systems to provide probabilistic translation features. Similarity features are very efficient to compute and therefore easier to use in dynamic adaptation than probabilistic features.

7.5.3 Qualitative evaluation

In this section, we analyse some concrete output examples that visualise the differences in the translations produced by the different models. We select examples from the TED portion of the test set because the BLEU scores indicated that both domain adaptation and topic adaptation improve translation on this set. Figure 7.2 shows two input and reference sentences with their translations under the unadapted baseline, the domain-adapted model and the model with both domain-adapted and topic-adapted features[14]. In the first example, the baseline translation is too short and does not translate the source verb *remontent* appropriately. This is fixed by the domain-adapted model and in addition, the topic-adapted model finds a contextually better translation that matches the reference. In the second example, the domain-adapted model fixes the wrong lexical choice of the baseline model and the topic-adapted model maintains the

[14]The outputs correspond to the models in the first line and the second block of Table 7.3.

Input	elles représentent les étendues de l'imagination humaine qui remontent à l'aube du temps.
BL	they represent the bodies of the human imagination back at the dawn of time.
+DA-TM	they represent the bodies of the human imagination **that** date **back to** the dawn of time.
+topics	they represent the bodies of the human imagination **that go back** the dawn of time.
Reference	they represent branches of the human imagination **that go back to** the dawn of time.
Input	ils l'ont fait en *drainant* les terres.
BL	they did in *drawing* the land.
+DA-TM	they did in **draining** the land.
+topics	they did in **draining** the land.
Reference	they did it by **draining** the land.

Figure 7.2: Comparison of translation output of different models: in these examples, domain adaptation yields most of the improvement in quality.

same translation. Thus, these are examples where domain adaptation is doing most of the adaptation work.

Figure 7.3 shows examples where all models make different lexical choices and only the addition of the topic-adapted model yields the correct lexical selection. In these examples, both the baseline and the domain-adapted model choose a translation that corresponds to a different sense of the French source word (*bitrate/throughput*, *settlement/agreement*), while the topic-adapted model selects a translation capturing the same sense as the reference translation (*flows*, *arrangement*)[15].

In order to draw the connection between the example outputs and the underlying features, Table 7.6 shows weighted feature scores for the three systems and phrase table features with high weights, which is $P(e|f)$ for the baseline and domain-adapted system. For the topic-adapted system, we show the feature scores of four translation features with the highest weights. For the baseline and domain-adapted system, the preference given by the weighted scores directly maps to the translation output in Figure 7.3. For the topic-adapted system, the probabilistic features prefer the translation from *débit* to *speed*, while the non-probabilistic features prefer the translation *flow*, which captures the sense of the reference translation. However, it seems that overall there is a stronger preference for *flow* since in both examples that translation appears

[15]Here we ignore the difference between the singular and plural of the word *flow*.

Input	le *débit* est en augmentation très rapide.
BL	the *speed* is growing very rapidly.
+DA-TM	the *throughput* is rising very fast.
+topics	the **flow** is growing very rapidly.
Reference	these **flows** are increasing very rapidly.
Input	le *débit* a augmenté.
BL	the *bitrate* has increased.
+DA-TM	the *throughput* has increased.
+topics	the **flow** has increased.
Reference	the **flows** have increased.

Figure 7.3: Comparison of translation output of different models: in these examples, topic adaptation yields the translation that captures the correct sense of the French source word.

débit →	**Baseline**	**Domain-adapted**		**Topic-adapted**		
speed	**0.830**	0.652	**0.924**	**0.890**	0.960	1.031
bitrate	**0.770**	0.606	0.873	0.831	0.918	1
throughput	0.700	**0.892**	0.874	0.889	0.919	1.026
flow	0.700	0.803	0.872	0.875	**0.979**	**1.058**

Table 7.6: Weighted feature scores of $P(e|f)$ for the baseline and domain-adapted system and the system with additional topic-adapted features (Conditional, Joint-Conditional, Sim-trgWord, TrgUnigrams). Scores for French source word *débit*.

in the output.

Finally, the examples in Figure 7.4 show an incremental improvement from the un-adapted model to the domain-adapted model and the topic-adapted model. In the first example, the baseline model produces a translation where the word order in the emphasized region is corrupted and the source word *répertoire* is translated in the wrong sense (*directory*). In the translation of the domain-adapted model, the lexical choice is corrected (*repertoire*) but the wrong word order still renders the text region incomprehensible. In the translation of the topic-adapted model, the correct word sense of *répertoire* is picked and the word order is also much improved, conveying the meaning that is expressed in the reference translation. In the second example, the domain-adapted model improves slightly over the baseline model by producing a more fluent translation. However, the underlined segments are still translated incorrectly, for example *historique de recherche* is translated as *record of my research*. The topic-adapted

Input	c'est une ayahuasca, dont beaucoup d'entre vous ont entendu parler, la plus puissante *préparation psychoactive du répertoire des shamans.*
BL	it's a ayahuasca , many of you have heard about it, the more powerful *shamans psychoactive preparation of the directory.*
+DA-TM	it's a ayahuasca, that many of you have heard about it, the more powerful *psychoactive shamans the* **repertoire** *of preparation.*
+topics	it's a ayahuasca, that many of you have heard about it, the more powerful *psychoactive preparation of the* **repertoire** *of shamans.*
Reference	this is ayahuasca, which many of you have heard about, the most powerful *psychoactive preparation of the shaman's* **repertoire**.
Input	et, si je veux m'éloigner et tout regarder je peux décortiquer *mon historique peut-être mon historique de recherche.*
BL	and, if i want to move me and look at everything i can go into *my historical historic perhaps my research.*
+DA-TM	and, if i want to get away from and look at everything i can go into *my maybe historical record of my research.*
+topics	and, if i want to get away from it and look at everything i can go into *my* **history** *can be my* **search history**.
Reference	and, if i want to step back and look at everything, i can slice and dice *my* **history** *perhaps by my* **search history**.

Figure 7.4: Comparison of translation output of different models: in these examples, we observe an incremental improvement from domain adaptation to topic adaptation.

model fixes the translations of the underlined segments and finds the correct translation *search history*.

The weighted feature scores for the first example of Table 7.4 are shown in Table 7.7. For this example, the preference according to all shown features matches the translation outcomes. The domain-adapted $P(e|f)$ as well as all topic-adapted features favour the correct translation *repertoire*.

All these examples show that domain and topic adaptation both contribute to the improved translation results and that depending on the input example, the contribution of one of the two models may be more important. The given examples do not allow for a definite conclusion on the relative strengths and weaknesses of the two models. Both models contributed to better lexical choice or more fluent translations, depending on the example. We therefore assume that the difference lies in the granularity of the modelled distributions rather than a clearly defined difference in the type of adaptation, such as style or genre versus topic.

répertoire →	**Baseline**	**Domain-adapted**		**Topic-adapted**		
directory	**0.931**	0.732	0.854	0.802	0.898	1
repertoire	0.831	**0.986**	**0.928**	**0.955**	**0.962**	**1.062**

Table 7.7: Weighted feature scores of $P(e|f)$ for the baseline and domain-adapted system and the system with additional topic-adapted features (Conditional, Joint-Conditional, Sim-trgWord, TrgUnigrams). Scores for French source word *répertoire*.

7.5.4 WADE evaluation

In this section we take a closer look at the results on the TED test set using the WADE framework. Since the BLEU and METEOR scores indicate that both domain and topic adaptation have an effect on the translation of TED documents, we want to investigate whether there is a qualitative difference between the effects of both types of adaptation. Table 7.8 shows the percentage of correct words out of all aligned word pairs in the source and reference sentences for the baseline system and the domain-adapted system, as well as the improvements when adding domain or topic adaptation on either of these systems. We distinguish output words according to whether they are content or function words and whether the distributions over target words have high or low entropy[16]. The latter is an indicator of how ambiguous the distributions are and thus whether contextual information would be expected to have an influence on lexical choice.

On the left side of Table 7.8, we observe that adding domain adaptation improves all four word categories but the improvements are larger for function and low entropy words. The opposite is true for adding topic adaptation, where the improvements are larger for content and high entropy words. On the right side of the table, we see that adding topic adaptation to a domain-adapted system has an even clearer effect, with a small negative or little gain for function and low entropy words as opposed to increases in % of correct content and high entropy words. While these results constitute only a preliminary analysis of more fine-grained word classes, they indicate that there may be a qualitative difference between domain adaptation and topic adaptation regarding the kind of improvements they yield over a baseline system. Further analysis is needed to verify whether these differences can be attributed to adaptation towards style and topic, respectively.

[16]High versus low entropy is determined in the same way as described in Section 4.6.8.

	Baseline	**+ LIN-TM**	**+ topics**	**LIN-TM**	**+topics**
	% correct	% improvement		% correct	% improvement
Content words	50.28	0.31	**0.34**	50.59	**0.28**
Function words	63.51	**0.39**	0.20	63.90	-0.05
High entropy words	44.81	0.34	**0.44**	45.15	**0.29**
Low entropy words	68.69	**0.41**	0.27	69.10	0.01

Table 7.8: Percentage of correctly translated words with the baseline system and a domain-adapted system (LIN-TM) and improvements over the respective baseline according to WADE. Source words are grouped by different sub-classes.

7.6 Conclusion

In this chapter, we have presented an approach to combining domain adaptation and topic adaptation within the same translation system, a subject which has not received any attention in the literature so far. We have analysed the relative benefit of both types of adaptation on a diverse set of test documents and found that the two approaches can be complementary depending on the text type and the level of overlap between their features sets.

An analysis of translation outputs has shown that both models contribute to the improved results and no clear distinction could be made between the kinds of improvements of each model. However, a more fine-grained analysis of the correctly translated words indicates that there may be a qualitative difference between the improvements yielded by domain and topic adaptation, with topic adaptation yielding larger gains on content and high entropy words. More analysis is needed to confirm whether this difference can be attributed to adaptation towards style and topic, respectively.

We have further shown that the domain of a test document can be predicted accurately by re-using trained topic models to build domain vector prototypes. Combining domain adaptation, topic adaptation and automatic domain prediction is useful when translating documents from unknown origin and can also help to reduce the load of test time computations while still benefitting from dynamic topic adaptation. Finding the right balance between both approaches such that a domain-adapted system provides adapted scores for the traditional phrase table features can lead to a more efficient architecture that combines online and offline computation. Our best combined model yields BLEU improvements of up to 1.67 over an unadapted baseline and 0.82 over a domain-adapted model.

Chapter 8

Conclusions

In this thesis, we have presented several new approaches to the problem of integrating contextual information from topic models into a statistical machine translation model.

We have shown that learning bilingual topic models over phrase pairs is feasible and improves translation performance over several domain adaptation baselines according to BLEU and METEOR. An intrinsic evaluation showed that our model, termed *Phrasal LDA*, learns useful topical structure that captures translation ambiguity which is confirmed by lower entropy and perplexity of the adapted distributions compared to the baseline distributions. We have also shown that the improvements in translation quality are stable across varying amounts of granularity for the latent topics. A comparison against topic-adapted baseline systems relying on monolingual topic models provides evidence that bilingual topic models are conceptually preferable to monolingual approaches and improve translation quality. Our work also includes the first direct comparison between SMT models with topic-adapted features and domain-adapted models.

We have explored an alternative way of introducing bilingual information into topic modelling by learning topic distributions over distributional profiles for pairs of source and target phrases. This model, termed *Phrase Pair Topic Model*, is conceptually different from our first model in that it does not explicitly represent the conditional relationship between source and target phrases. Instead, it follows a distributional approach to meaning representation which simplifies the selection of semantically appropriate target phrases for a given test context. We have also shown that this model allows us to combine information from sentence-level and document-level context and that this benefits translation quality.

For both proposed models, we find that a combination of several topic-adapted

features yields better results than adapting or adding just a single translation feature. This can be explained by the fact that the bias of the baseline model towards particular translations has to be overcome by the adapted features.

Finally, we have explored the relationship between domain adaptation and topic adaptation by combining features of both adaptation types in the same log-linear model. Since our test documents are not assumed to have domain labels, this involved automatically predicting the domain of a test document using topic-based classifiers. We have found that the two approaches can be complementary, depending on the text type and the level of overlap between their feature sets. We argue that combining both techniques could help to build more efficient translation systems that combine offline and online adaptation.

Preliminary evaluations of translation quality using the WADE framework indicate that there is a qualitative difference in the improvements in translation quality achieved by domain adaptation and topic adaptation methods. Comparisons of several domain-adapted and topic-adapted models indicate that domain adaptation yields larger gains on function words and low entropy words while topic adaptation yields larger gains on content words and high entropy words. This could be an indicator that domain adaptation is better at adapting to style, which is expected to have an effect on function words and auxiliary verbs (which we count as function words), while topic adaptation is better at translating ambiguous words, enabled by adaptation to more fine-grained context. However, further experimentation is needed to validate this hypothesis.

8.1 Comparison of proposed models

Comparing our different models against each other, the *Phrasal LDA* model yields the best results overall, with and without domain adaptation on top of topic adaptation. The differences in evaluation scores are relatively small but the *Phrasal LDA* model uses fewer adapted features to achieve these results. For example, combined with a domain-adapted language model it yields an overall BLEU score of 27.84 while the *Phrase Pair Topic Model* (with extended feature set) combined with a domain-adapted language model yields a BLEU score of 27.60. One reason for the slightly superior results might be that there is a conceptual difference between the models. The *Phrasal LDA* model adapts to the test context by computing posterior distributions at the phrase level, which means that there is a different underlying topic distribution for each source phrase in the test document. This lets the model take into account differences between

phrase pairs consisting only of function words, which are typically less dependent on the context, and phrase pairs containing ambiguous content words. The *Phrase Pair Topic Model* learns topic distributions for each phrase pair at training time and compares them with the topic distribution of the test context. It is possible that this approach is at a disadvantage because of its coarser representation of the test context in the adaptation step. Even though we tried to equip the *Phrase Pair Topic Model* model with similar features as the *Phrasal LDA* model, the models are very different in training and it may be the case that the *Phrasal LDA* model learns better distributions at training time.

8.2 Limitations of this work

Most of the experiments presented in this thesis have been carried out using relatively small training data sets, by MT standards. One of the reasons is that depending on the number of topics the inference in the bilingual *phrasal LDA* model proved to take quite a long time to converge to reasonable results. Therefore, we limited the training data to a size that was manageable to train to convergence and at the same time was large enough to train a realistic baseline model. We expect that similar effects of overlapping translation distributions hold for small or large data sets of similar diversity. Another reason is that the test environment was set up in a way to simulate data diversity in training, development and test sets without any particular bias for one of the corpora. Here we were guided by the size of the TED corpus and selected equal amounts of data from other corpora. Our test set contains documents that are more diverse than the test sets of standard MT evaluations such as the Workshop for Machine Translation (Bojar et al., 2014), which consist exclusively of news articles.

In the future, we will aim for data setups with larger amounts of training data while maintaining development and test sets from diverse domains. It is likely that our topic models could be trained on subsets of the training data and still learn enough about the topical structure to improve translation. We will explore this possibility in the future in order to scale our techniques up to large training sets.

8.3 Future work

There are many possible extensions to the models presented in this thesis and in the following we point out some directions.

8.3.1 Integration of domain knowledge into topic modelling

Even though we have shown that our models are able to outperform supervised domain adaptation methods, we have also noticed that domain adaptation methods can be complementary to topic modelling approaches. A very useful extension to our models would therefore be to directly incorporate domain information as prior knowledge during training. For example, it would be straightforward to incorporate a hierarchical prior into the *phrasal LDA* model and this prior could be made dependent on domain knowledge. If we treat domain labels as random variables just like topics (though domains are observed in training), we can similarly infer the domain of a document at test time. Such a model could provide the ability to learn topics that are more fine-grained than the training domains but fall back to domain-adapted probabilities when there is no evidence for more structure.

8.3.2 Adaptation as modulation in selection preference

Another avenue for future work could be to distinguish between semantic and syntactic aspects in the local context. While it is reasonable to assume that many content words are influenced by the overall topic of a document, there are other dependencies that could be considered. For example, there is the notion of selectional preference between nouns and adjectives as well as between predicates and their arguments. One could view the topic changes in dynamic adaptation as changes in the selectional preference of word categories. This would allow us to adapt the translation probabilities of some words directly while the translation of other words would be dependent on the preference of their governing word. For example, the translation of an adjective would only depend on the preference of its associated noun, as in the French source phrase *un petit editeur* which can translate either to *lightweight editor* or to *small publisher* in English. Here, the translation of the noun depends on the context, but the translation of the adjective depends primarily on the sense of the noun.

8.3.3 Topic adaptation in a semi-automated translation scenario

An interesting application of our topic modelling approaches would be to integrate them in a translation scenario where a human translator either post-edits an automatic translation draft or manually combines translation options proposed by a machine translation system. While the source text under translation is available ahead of

time, the translations or translation post-edits become available incrementally. The machine translation system could learn from each new translation after it is completed and before translating or suggesting options for the next sentence. Since the topic model learns associations between target phrases and topics, the translations successively becoming available to the system would provide additional, potentially corrective information about the topical context in which the translation is taking place. Therefore, topic inference on the source side could be complemented with additional adaptation steps that treat the new target sentences as additional (training) data to refine the context topic mixture. Because of the close relationship between training and test inference, integration of these additional adaptation steps would be easy to realise.

8.4 Final remarks

In the last two years, there has been a rapid development of neural network modelling techniques for SMT (Schwenk, 2012; Le et al., 2012; Auli et al., 2013; Kalchbrenner and Blunsom, 2013; Devlin et al., 2014). All of these models make use of contextual information from the source or target sentence and some take longer histories into account by introducing recurrency (Auli et al., 2013; Kalchbrenner and Blunsom, 2013). Devlin et al. (2014) have shown large BLEU improvements and it is likely that neural network-based models will be increasingly used for machine translation in the future.

Thus far, these models have not been evaluated in domain or topic adaptation scenarios and therefore it is difficult to say to what extent the improvements from neural networks overlap with the improvements achieved by domain or topic adaptation techniques. Such comparisons would be very interesting and would help in deciding whether neural network approaches and topic modelling approaches to contextual adaptation can fill complementary roles. To the best our knowledge, the work of Cui et al. (2014) is the only work that directly compares a neural network-based approach to a topic modelling approach. Cui et al. show small gains from replacing topic models with neural networks for adaptation, but they also enriched the source side information with additional retrieved documents. As a consequence, it is difficult to decide whether the improvements are due to this additional information or to the modelling power of the neural networks.

It would be relatively straightforward to use a neural network instead of a topic model in the *Phrase Pair Topic Model*, since we could equally apply cosine similarity to the latent representations learned by a neural network. In fact, the approach would

then be quite similar to the work of Cui et al. (2014). Like the work of Xiao et al. (2012), their model uses hierarchical translation rules and a so-called sensitivity feature that boosts translation rules with high-entropy topic distributions which are deemed general enough to be applied in all topical contexts. Because the hierarchical system contains purely lexicalised rules as well as more general structural rules, there is the potential to learn which kind of syntactic constructions are generally useful and which only occur in specific topical contexts. It would be interesting to investigate whether a hierarchical system benefits more from topic adaptation because of the distinction between lexicalised and structural rules.

We believe that the increased activity in research on contextual modelling for MT demands for better comparative evaluation to determine the nature of the improvements in translation quality. Being able to distinguish the qualitative differences between translation models can lead to a better understanding of how to extend and combine existing models. The use of test sets that represent a variety of text genres and topics and a more focused evaluation of the translation output, as carried out in this thesis, could be a step in this direction.

Appendix A

Additional Material for Chapter 4

Model	Mixed	CC	NC	TED
ALL	26.72	20.77	29.91	30.13
lex(e|f,d)	26.84	20.97	29.94	30.25
trgUnigrams	26.84	20.94	29.97	30.27
docSim	26.92	21.02	30.05	30.33
p(e|f,d)	26.92	21.06	30.00	30.34
All features	**27.13**	**21.23**	**30.24**	**30.59**

Table A.1: METEOR scores of pLDA features (50 topics), separately and combined.

Model	Mixed	CC	NC	TED
ALL	26.72	20.77	29.91	30.13
3 topics	26.75	20.88	29.87	30.15
4 topics	26.90	20.94	30.15	30.25
5 topics	27.00	20.96	30.15	30.55
10 topics	27.12	21.20	30.22	**30.59**
20 topics	27.13	21.26	30.21	30.57
50 topics	27.13	21.23	**30.24**	**30.59**
100 topics	**27.15**	**21.35**	30.23	30.49
>ALL	+0.43	+0.58	+0.33	+0.46

Table A.2: METEOR scores of baseline and topic-adapted systems (pLDA) with all 4 features.

Data	Mixed	CC	NC	TED
FILLUP	26.81	20.63	30.09	30.38
LIN-TM	26.87	20.76	30.10	30.43
pLDA	**27.15**	**21.35**	**30.23**	**30.49**
>FILLUP	+0.34	+0.72	+0.14	+0.11
>LIN-TM	+0.28	+0.59	+0.13	+0.06

Table A.3: Comparison of best pLDA system with two domain-aware benchmark systems, according to METEOR.

Model	Mixed	CC	NC	TED
LIN-LM				
+ ALL	26.83	20.81	30.09	30.22
+ FILLUP	26.82	20.60	30.14	30.37
+ LIN-TM	26.87	20.73	30.15	30.39
+ pLDA	**27.17**	**21.31**	**30.22**	**30.64**
>ALL	+0.34	+0.50	+0.13	+0.42

Table A.4: Combination of all models with additional LM adaptation (pLDA: 50 topics), according to METEOR.

noyau→nucleus:
0.311531 0.193719 0.000169 0.002358 0.028055
0.062241 0.026591 0.006102 0.005119 **0.263657**
0.008763 0.010780 0.018648 0.016688 0.001060
0.009532 0.009338 0.000150 0.002791 0.022706
noyau→core:
0.232056 0.001337 0.000790 0.005910 0.023344
0.007239 0.002242 0.004521 0.148810 0.003026
0.013628 **0.501909** 0.006114 0.018770 0.001372
0.014654 0.002821 0.000961 0.008630 0.001866
noyau→kernel:
0.255461 0.018751 0.003026 0.045590 0.017920
0.019983 0.003934 0.020499 0.001524 0.006208
0.007618 0.002757 0.026623 0.004274 0.002117
0.002761 0.006954 0.000667 0.004771 **0.548563**
flux→stream:
0.246312 0.019178 0.001046 0.004973 0.005710
0.010538 0.003247 0.005004 0.004664 0.005454
0.001898 0.001868 0.010821 0.001862 0.003114
0.002486 0.006646 0.000521 0.000828 **0.663833**
alteration→impairment:
0.286249 0.000135 0.049673 0.000105 0.085986
0.000257 0.008845 0.000279 **0.182224 0.312469**
0.000127 0.000094 0.000183 0.000185 0.000144
0.000380 0.000351 0.000697 0.071518 0.000099
acolytes→acolytes:
0.344976 0.070804 0.001061 0.005074 0.048539
0.081214 0.106070 0.010384 <u>0.013203</u> 0.134071
<u>0.037854</u> 0.006507 0.045148 0.025617 0.004940
0.048680 0.003187 0.000952 0.009804 0.001915
propension→propensity:
0.258960 0.003896 0.002498 0.011905 0.118790
0.004922 0.001901 0.002170 0.180199 <u>0.010836</u>
0.037763 <u>0.074378</u> 0.023212 0.081500 0.003883
0.115101 0.002185 0.000475 0.057304 0.008120

Figure A.1: Document topic distributions of translation examples in Section 4.6.4. Each block is denoted by the source word in question and its correct translation in that document. Topic proportions are marked in bold when they correspond to a mode in the distribution where the correct translation is likely under the respective topic (see topic-specific translation tables) and underlined if there is no mode at the appropriate topic.

flux		
topic 8	flows = 0.54	inflows = 0.15
topic 9	streamlines = 0.29	fluctuation = 0.22
topic 11	flow* = 0.44	flows = 0.28
topic 19	stream = 0.46	feed = 0.07
altération		
topic 8	impairment* = 0.40	corruption = 0.36
topic 9	impairment* = 0.95	corruption = 0.03
topic 14	alteration* = 0.75	impairment = 0.22
topic 19	corruption* = 0.46	alteration = 0.42

Table A.5: The two most probable translations of the French source words *flux* and *altération* and their probabilities under different latent topics (*: preferred by ALL). Some representative topics according to manual inspection are 8: politics, 9: science, 11: economy, 19: IT.

acolytes		
topic 6	henchmen = 0.92	cohorts = 0.06
topic 8	acolytes* = 0.69	cheerleaders = 0.05
topic 9	cohorts = 0.48	henchmen = 0.23
topic 10	acolytes* = 0.45	cheerleaders = 0.23
topic 15	cronies = 0.36	cheerleaders = 0.35
propension		
topic 4	tendency = 0.28	readiness = 0.24
topic 8	willingness = 0.15	propensity = 0.08
topic 9	propensity* = 0.83	predilection = 0.15
topic 11	propensity* = 0.50	willingness = 0.08

Table A.6: The two most probable translations of the French source words *acolytes* and *propension* and their probabilities under different latent topics (*: preferred by ALL).

	Tokens	**Types**
All words	54517	7021
Content words	25188	6674
Function words	29329	370
High entropy words	15908	2643
Low entropy words	29719	1057

Table A.7: Number of aligned word tokens and types in WADE computations for TED portion of test set. Note that the content and function word tokens add up the the total number of word tokens while the sum of high and low entropy word tokens contains only those tokens that have a single-word entry in the baseline phrase table.

Appendix B

Additional Material for Chapter 5

System	English-Spanish		French-English		Dutch-English	
	best	oof	best	oof	best	oof
Baseline	0.674	0.854	**0.722**	0.884	0.613	0.750
50-topics	**0.682**	0.860	0.719	**0.896**	0.616	**0.759**
mixture:geoAvg	0.677	**0.863**	0.715	**0.896**	**0.619**	0.756
mixture:max	0.679	0.860	0.712	0.887	0.618	0.753

Table B.1: Results for SemEval 2014, Task 5: Word accuracy (best and out-of-five) of the baseline system and the systems with added context similarity feature. All systems were run without scoring the language model context.

System	English-Spanish		French-English		Dutch-English	
	best	oof	best	oof	best	oof
Baseline + LM context	**0.839**	0.944	0.823	0.934	0.686	0.809
run1: 50-topics + LM context	0.827	0.946	**0.824**	0.938	**0.692**	**0.811**
run2: mixture:geoAvg + LM context	0.827	0.944	0.821	**0.939**	0.688	0.808
run3: mixture:max + LM context	0.820	**0.949**	0.816	0.937	0.688	0.808
2nd-ranked systems	0.809[1]	0.887[2]	0.694[2]	0.839[2]	0.679[3]	0.753[3]

Table B.2: Results for SemEval 2014, Task 5: Word accuracy (best and out-of-five) of all submitted systems (runs 1-3) as well as the baseline system without the context similarity feature. All systems were run with the language model context provided via XML input. Systems on 2nd rank: [1]UNAL-run2, [2]CNRC-run1, [3]IUCL-run1.

Appendix C

Additional Material for Chapter 6

Type of adaptation	Model	Mixed	CC	NC	TED
Domain-adapted	LIN-TM	26.87	20.76	30.10	30.43
	FILLUP	26.81	20.63	30.09	30.38
Topic-adapted	global⊕local	27.06	21.11	30.24	30.49
	>LIN-TM	+0.19	+0.35	+0.14	+0.06
	>FILLUP	+0.25	+0.48	+0.15	+0.11

Table C.1: METEOR scores of translation model using similarity features derived from PPT model (50 topics) in comparison with two (supervised) domain-adapted systems.

Model	Mixed	CC	NC	TED
Baseline	26.72	20.77	29.91	30.13
+ docSim	26.92	21.02	30.05	30.33
+ phrSim-global ⊕ phrSim-local	27.12	**21.20**	**30.26**	30.57
+ phrSim-global ⊛ phrSim-local	**27.13**	21.19	**30.26**	**30.60**
global⊛local>BL	+0.41	+0.42	+0.35	+0.47

Table C.2: METEOR scores of baseline, baseline + document similarity feature and additional phrase pair similarity features. All models were trained with 50 topics.

Appendix D

Additional Material for Chapter 7

Model	Mixed	CC	NC	TED
Baseline	26.86	19.61	29.42	31.88
+ Conditional	27.34	**20.37**	29.41	32.48
+ Joint-Conditional	27.23	20.04	29.51	32.52
+ Conditional	**27.36**	20.20	**29.74**	**32.62**

Table D.1: BLEU scores of baseline system with the addition of the topic-adapted features *Conditional* and *Joint-Conditional*, separately or combined.

Model	Mixed	CC	NC	TED
Baseline	26.72	20.77	29.91	30.13
+ topics	27.06	21.26	30.02	30.59
	+0.34	+0.49	+0.11	+0.46
DA-TM	26.87	20.76	30.10	30.43
+ topics	**27.11**	**21.24**	**30.11**	**30.68**
	+0.24	+0.48	+0.01	+0.25
DA-LM	26.83	20.81	30.09	30.22
+ topics	27.06	21.21	30.10	30.55
	+0.23	+0.40	+0.01	+0.33
DA-TM+LM	26.87	20.73	30.15	30.39
+ topics	27.06	21.13	30.14	30.60
	+0.19	+0.40	-0.01	+0.21
Gain of best system over baseline	+0.39	+0.47	+0.20	+0.55

Table D.2: METEOR results of unadapted/adapted baseline models and additional topic-adapted features (Overlap) with their gain over the respective baseline (bottom of each box). The best system on the mixed test set is marked in bold.

Model	**Mixed**	**CC**	**NC**	**TED**
Baseline	26.72	20.77	29.91	30.13
+ TrgUnigrams	26.76	20.92	29.78	30.24
+ DA-TM	26.98	20.94	30.12	30.59
+ Sim-phrasePair	26.99	21.10	30.07	30.46
+ DA-TM	27.01	20.95	30.22	30.53
+ Sim-targetPhrase	26.91	20.98	30.04	30.35
+ DA-TM	26.99	20.93	30.15	30.58
+ Sim-targetWord	26.78	20.86	29.90	30.25
+ DA-TM	26.96	20.88	30.15	30.51
+ Sim-combine	26.92	21.02	30.01	30.39
+ DA-TM	**27.08**	**21.02**	**30.25**	**30.66**
+ Sim-combine-loglin	26.88	21.01	29.97	30.29
+ DA-TM	26.97	20.92	30.14	30.55
+ Sim-combine+trgUnigrams	26.86	21.00	29.89	30.38
+ DA-TM	26.97	20.93	30.08	30.62
DA gain of best system	+0.16	+0.00	+0.24	+0.27
Gain of best system over baseline	+0.36	+0.25	+0.34	+0.53

Table D.3: METEOR results of smaller topic feature sets with added domain-adapted features. The best system on the mixed test set is marked in bold.

Bibliography

Agirre, E., Lopez de la Calle, O., and Martinez, D. (2005a). Exploring feature set combinations for WSD. In *Proceedings of the SEPLN*, pages 285–291.

Agirre, E. and Lopez De Lacalle, O. (2008). On Robustness and Domain Adaptation using SVD for Word Sense Disambiguation. In *Proceedings of Coling*, pages 17–24.

Agirre, E., Lopez de Lacalle, O., and Martınez, D. (2005b). Exploring feature spaces with SVD and unlabeled data for Word Sense Disambiguation. In *Proceedings of the Conference on Recent Advances on Natural Language Processing*.

Andrzejewski, D., Zhu, X., and Craven, M. (2009). Incorporating Domain Knowledge into Topic Modeling via Dirichlet Forest Priors. *Proceedings of the International Conference on Machine Learning*, pages 25–32.

Apidianaki, M., Wisniewski, G., and Sokolov, A. (2012). WSD for n-best reranking and local language modeling in SMT. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 1–9.

Auli, M., Galley, M., Quirk, C., and Zweig, G. (2013). Joint Language and Translation Modeling with Recurrent Neural Networks. In *Proceedings of EMNLP*, number October, pages 1044–1054.

Axelrod, A., He, X., and Deng, L. (2012). New methods and evaluation experiments on translating TED talks in the IWSLT benchmark. *Proceedings of ICASSP*.

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of EMNLP*.

Banchs, R. E. and Costa-jussà, M. R. (2011). A Semantic Feature for Statistical Machine Translation. In *SSST-5 Proceedings of the Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Banerjee, P., Du, J., Li, B., Naskar, S. K., Way, A., and Genabith, J. V. (2010). Combining Multi-Domain Statistical Machine Translation Models using Automatic Classifiers. In *Proceedings of AMTA*.

Banerjee, S. and Lavie, A. (2005). METEOR : An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*.

Bansal, M., Denero, J., and Lin, D. (2012). Unsupervised Translation Sense Clustering. In *Proceedings of NAACL HLT*, pages 773–782.

Bertoldi, N. and Federico, M. (2009). Domain adaptation for statistical machine translation with monolingual resources. *Proceedings of the Fourth Workshop on Statistical Machine Translation*.

Bisazza, A., Ruiz, N., and Federico, M. (2011). Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation. In *Proceedings of IWSLT*.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc.

Blei, D., Ng, A., and Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Proceedings of WMT 2013*.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-amand, H., Soricut, R., Specia, L., and Tamchyna, A. (2014). Findings of the 2014 Workshop on Statistical Machine Translation. pages 12–58.

Boyd-Graber, J. and Blei, D. (2009). Multilingual Topic Models for Unaligned Text. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Boyd-Graber, J., Blei, D., and Zhu, X. (2007). A Topic Model for Word Sense Disambiguation. In *Proceedings of EMNLP-CoNLL*, pages 1024–1033.

Boyd-Graber, J. and Resnik, P. (2010). Holistic Sentiment Analysis Across Languages: Multilingual Supervised Latent Dirichlet Allocation University of Maryland. In *Proceedings of EMNLP*, pages 45–55.

Brown, P. F., Cocke, J., Pietra, S. A. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

Cai, J. F., Lee, W. S., and Teh, Y. W. (2007). Improving Word Sense Disambiguation Using Topic Features. In *Proceedings of EMNLP*, pages 1015–1023.

Carpuat, M. (2009). One Translation per Discourse. In *Proceedings of the NAACL HLT Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 19–27.

Carpuat, M. and Wu, D. (2005). Word Sense Disambiguation vs . Statistical Machine Translation. In *Proceedings of ACL*, pages 387–394.

Carpuat, M. and Wu, D. (2007a). Context-Dependent Phrasal Translation Lexicons for Statistical Machine Translation. In *Proceedings of MT Summit*.

Carpuat, M. and Wu, D. (2007b). How Phrase Sense Disambiguation outperforms Word Sense Disambiguation for Statistical Machine Translation. In *International Conference on Theoretical and Methodological Issues in MT*.

Carpuat, M. and Wu, D. (2007c). Improving Statistical Machine Translation using Word Sense Disambiguation. In *Proceedings of EMNLP*, pages 61–72.

Cettolo, M., Girardi, C., and Federico, M. (2012). Witˆ3: Web inventory of transcribed and translated talks. In *Proceedings of EAMT*.

Chan, Y. S., Ng, H. T., and Chiang, D. (2007). Word Sense Disambiguation Improves Statistical Machine Translation. In *Proceedings of ACL*.

Chen, B., Foster, G., and Kuhn, R. (2013a). Adaptation of Reordering Models for Statistical Machine Translation. In *Proceedings of NAACL*, pages 938–946.

Chen, B., Foster, G., Kuhn, R., Boulevard, A.-t., Québec, G., and Jx, C. (2010). Bilingual Sense Similarity for Statistical Machine Translation. In *Proceedings of ACL*, pages 834–843.

Chen, B., Kuhn, R., and Foster, G. (2013b). Vector Space Model for Adaptation in Statistical Machine Translation. In *Proceedings of ACL*, pages 1285–1293.

Cherry, C. and Foster, G. (2012). Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of NAACL*.

Chew, P. A., Verzi, S. J., Bauer, T. L., and Mcclain, J. T. (2006). Evaluation of the Bible as a Resource for Cross-Language Information Retrieval. In *Proceedings of the Workshop on Multilingual Language Resources and Interoperability*, pages 68–74.

Chiang, D. (2007). Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228.

Chiang, D. (2012). Hope and fear for discriminative training of statistical translation. *Journal of Machine Learning Research*, 13.

Chiang, D., Knight, K., and Wang, W. (2009). 11,001 New Features for Statistical Machine Translation. In *Proceedings of HLT-NAACL*.

Chiang, D., Marton, Y., and Resnik, P. (2008). Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP 08*, number October, page 224. Association for Computational Linguistics.

Collins, M. (2002). Discriminative Training Methods For Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of EMNLP*, pages 1–8.

Costa-jussà, M. R. and Banchs, R. E. (2010). A vector-space dynamic feature for phrase-based statistical machine translation. *Journal of Intelligent Information Systems*, 37(2):139–154.

Crammer, K. and Singer, Y. (2003). Ultraconservative Online Algorithms for Multiclass Problems. *Journal of Machine Learning Research*, 3(4-5):951–991.

Cui, L., Chen, X., Zhang, D., Liu, S., Li, M., and Zhou, M. (2013). Multi-domain Adaptation for SMT Using Multi-task Learning. In *Proceedings of EMNLP*, pages 1055–1065.

Cui, L., Zhang, D., Liu, S., Chen, Q., Li, M., Zhou, M., and Yang, M. (2014). Learning Topic Representation for SMT with Neural Networks. In *Proceedings of ACL*.

Daumé III, H. and Jagarlamudi, J. (2011). Domain Adaptation for Machine Translation by Mining Unseen Words University of Maryland. In *Proceedings of ACL*, pages 407–412.

Deerwester, S., Dumais, S., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *Proceedings of ACL*, pages 1370–1380.

Dinu, G. and Lapata, M. (2010). Measuring Distributional Similarity in Context. In *Proceedings of EMNLP*, pages 1162–1172.

Eidelman, V., Boyd-Graber, J., and Resnik, P. (2012). Topic Models for Dynamic Translation Model Adaptation. In *Proceedings of ACL*.

Eisele, A. and Chen, Y. . (2010). MultiUN: A multilingual corpus from united nation documents. In *Proceedings of LREC*.

Finch, A. (2008). Dynamic Model Interpolation for Statistical Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 208–215.

Firth, J. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis (special volume of the Philological Society)*, 1952-1959:1–32.

Foster, G., Goutte, C., and Kuhn, R. (2010a). Discriminative Instance Weighting for Domain Adaptation in Statistical Machine Translation. In *Proceedings of EMNLP*.

Foster, G., Isabelle, P., Kuhn, R., and Canada, C. (2010b). Translating Structured Documents. In *Proceedings of AMTA*.

Foster, G. and Kuhn, R. (2007). Mixture-Model Adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*.

Gale, W. A., Church, K. W., and Yarowsky, D. (1992). One Sense Per Discourse. In *Proceedings of the workshop on Speech and Natural Language*.

Gao, J., He, X., Yih, W.-t., and Deng, L. (2013). Learning Semantic Representations for the Phrase Translation Model. Technical report.

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Gimenez, J. and Marquez, L. (2007). Context-aware Discriminative Phrase Selection for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 159–166.

Gimpel, K. and Smith, N. A. (2008). Rich Source-Side Context for Statistical Machine Translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 9–17.

Gliozzo, A., Giuliano, C., and Strapparava, C. (2005). Domain Kernels for Word Sense Disambiguation. In *Proceedings of ACL*, pages 403–410.

Gong, Z., Zhang, M., and Guodong, Z. (2011). Cache-based Document-Level Statistical Machine Translation. In *Proceedings of EMNLP 2011*.

Gong, Z., Zhang, Y., and Zhou, G. (2010). Statistical Machine Translation based on LDA. In *4th International Universal Communication Symposium (IUCS)*.

Gong, Z. and Zhou, G. (2011). Employing topic modeling for statistical machine translation. In *Proceedings of IEEE 2011*.

Green, S., Wang, S., Cer, D., and Manning, C. D. (2013). Fast and Adaptive Online Training of Feature-Rich Translation Models. In *Proceedings of ACL*.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101 Suppl:5228–35.

Gruber, A., Rosen-Zvi, M., and Weiss, Y. (2007). Hidden Topic Markov Models. In *Journal of Machine Learning Research*.

Haddow, B. and Koehn, P. (2012). Analysing the Effect of Out-of-Domain Data on SMT Systems. In *Proceedings of WMT*.

Hardmeier, C. and Nivre, J. (2012). Document-Wide Decoding for Phrase-Based Statistical Machine Translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1179–1190.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

Hasan, S., Ganitkevitch, J., Ney, H., and Technology, H. L. (2008). Triplet Lexicon Models for Statistical Machine Translation. In *Proceedings of EMNLP*, pages 372–381.

Hasler, E. (2014). UEdin: Translating L1 phrases in L2 context using context-sensitive SMT. In *Proceedings of the 8th International Workshop on Semantic Evaluation*.

Hasler, E., Bell, P., Ghoshal, A., Haddow, B., Koehn, P., Mcinnes, F., Renals, S., and Swietojanski, P. (2012a). The UEDIN Systems for the IWSLT 2012 Evaluation. In *Proceedings of the IWSLT*.

Hasler, E., Blunsom, P., Koehn, P., and Haddow, B. (2014a). Dynamic Topic Adaptation for Phrase-based MT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Hasler, E., Haddow, B., and Koehn, P. (2011). Margin Infused Relaxed Algorithm for Moses. *The Prague Bulletin of Mathematical Linguistics No. 96*.

Hasler, E., Haddow, B., and Koehn, P. (2012b). Sparse Lexicalised Features and Topic Adaptation for SMT. In *Proceedings of IWSLT*.

Hasler, E., Haddow, B., and Koehn, P. (2014b). Combining Domain and Topic Adaptation for SMT. In *Proceedings of AMTA*.

Hasler, E., Haddow, B., and Koehn, P. (2014c). Dynamic Topic Adaptation for SMT using Distributional Profiles. In *Proceedings of the 9th Workshop on Statistical Machine Translation*.

He, Z., Liu, Q., and Lin, S. (2008). Improving Statistical Machine Translation using Lexicalized Rule Selection. In *Proceedings of Coling*, pages 321–328.

Heinrich, G. (2009). Parameter estimation for text analysis. Technical report.

Hewavitharana, S., Mehay, D. N., and Ananthakrishnan, S. (2013). Incremental Topic-Based Translation Model Adaptation for Conversational Spoken Language Translation. In *Proceedings of ACL*, pages 697–701.

Hildebrand, A. S., Eck, M., Vogel, S., and Waibel, A. (2005). Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval. In *Proceedings of EAMT*, pages 133–142.

Hofmann, T. (1999). Probabilistic Latent Semantic Analysis. In *Proceedings of Uncertainty in Artificial Intelligence*.

Hopkins, M. and May, J. (2011). Tuning as ranking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Hsu, B.-J. P. and Glass, J. (2006). Style & topic language model adaptation using HMM-LDA. In *Proceedings of the EMNLP*, page 373.

Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. (2012). Improving Word Representations via Global Context and Multiple Word Prototypes. In *Proceedings of ACL*, pages 873–882.

Irvine, A., Morgan, J., Carpuat, M., III, H. D., and Munteanu, D. (2013). Measuring Machine Translation Errors in New Domains. In *Proceedings of TACL*.

Ittycheriah, A. and Roukos, S. (2007). Direct Translation Model 2. In *Proceedings of NAACL HLT*, pages 57–64.

Joshi, M., Pedersen, T., and Maclin, R. (2005). A Comparative Study of Support Vector Machines Applied to the Supervised Word Sense Disambiguation Problem in the Medical Domain. In *Proceedings of IICAI,*, pages 3449–3468.

Jurafsky, D. and Martin, J. (2008). *Speech and Language Processing (2nd Edition)*. Prentice Hall, 2 edition.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent Continuous Translation Models. In *Proceedings of EMNLP*.

Koehn, P. (2004a). Pharaoh: A Beam Search Decoder For Phrase-Based Statistical Machine Translation Models. In *Proceedings of AMTA*.

Koehn, P. (2004b). Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP*.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for SMT. In *Proceedings of ACL: Demo and poster sessions*.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of NAACL*, pages 48–54.

Koehn, P. and Schroeder, J. (2007). Experiments in Domain Adaptation for Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227.

Lau, J. H., Cook, P., Mccarthy, D., Newman, D., and Baldwin, T. (2010). Word Sense Induction for Novel Sense Detection. In *Proceedings of EACL*.

Le, H. S., Allauzen, A., and Yvon, F. (2012). Continuous Space Translation Models with Neural Networks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Lefever, E. and Hoste, V. (2013). SemEval-2013 Task 10: Cross-lingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation, in Conjunction with the Second Joint Conference on Lexical and Computational Semantics*.

Li, L., Roth, B., and Sporleder, C. (2010). Topic Models for Word Sense Disambiguation and Token-based Idiom Detection. In *Proceedings of ACL*, pages 1138–1147.

Liang, P., Bouchard-Côté, A., Klein, D., and Taskar, B. (2006). An end-to-end discriminative approach to machine translation. In *Proceedings of ACL*, pages 761–768.

Louis, A. and Webber, B. (2014). Structured and Unstructured Cache Models for SMT Domain Adaptation. In *Proceedings of EACL*, pages 155–163.

Mackay, D. J. C. (2003). *Information Theory , Inference , and Learning Algorithms*.

Matsoukas, S., Rosti, A.-V. I., and Zhang, B. (2009). Discriminative corpus weight estimation for machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Mauser, A., Hasan, S., and Ney, H. (2009). Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Proceedings of EMNLP*.

McCallum, A. K. (2002). MALLET: A Machine Learning for Language Toolkit.

McDonald, R., Crammer, K., and Pereira, F. (2005). Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, number June, pages 91–98.

Mei, Y. (2010). Contextual Modeling for Meeting Translation Using Unsupervised Word Sense Disambiguation. In *Proceedings of Coling*, number August, pages 1227–1235.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087.

Minka, T. P. (2012). Estimating a Dirichlet distribution. Technical Report 8.

Mohammad, S. and Hirst, G. (2006). Distributional Measures of Concept-Distance :. In *Proceedings of EMNLP*.

Moore, R. C. and Lewis, W. (2010). Intelligent Selection of Language Model Training Data. In *Proceedings of ACL*, pages 220–224.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. *Proceedings of the 41st Annual Meeting on Association*, pages:160.

Och, F. J. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, number July.

Och, F. J. and Ney, H. (2004). The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4).

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of ACL*, pages 311–318.

Pecina, P., Toral, A., Way, A., Papavassiliou, V., Prokopidis, P., and Giagkou, M. (2011). Towards Using Web-Crawled Data for Domain Adaptation in Statistical Machine Translation. In *Proceedings of EAMT*.

Razmara, M. (2012). Mixing Multiple Translation Models in Statistical Machine Translation. In *Proceedings of ACL*.

Ruiz, N. and Federico, M. (2012). MDI Adaptation for the Lazy: Avoiding Normalization in LM Adaptation for Lecture Translation. In *Proceedings of IWSLT 2012*.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees.

Schwenk, H. (2008). Investigations on Large-Scale Lightly-Supervised Training for Statistical Machine Translation. In *Proceedings of IWSLT*, pages 182–189.

Schwenk, H. (2012). Continuous Space Translation Models for Phrase-Based Statistical Machine Translation. In *Proceedings of Coling*, number December 2012, pages 1071–1080.

Schwenk, H. and Koehn, P. (2008). Large and Diverse Language Models for Statistical Machine Translation. In *Proceedings of IJCNLP*, pages 661–666.

Sennrich, R. (2012a). Mixture-modeling with unsupervised clusters for domain adaptation in statistical machine translation. In *Proceedings of EAMT*, pages 185–192.

Sennrich, R. (2012b). Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation. In *Proceedings of EACL*.

Sennrich, R., Schwenk, H., and Aransa, W. (2013). A Multi-Domain Translation Model Framework for Statistical Machine Translation. In *Proceedings of ACL*, pages 832–840.

Shannon, C. E. (1948). A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423.

Shen, L., Xu, J., Zhang, B., Matsoukas, S., and Weischedel, R. (2009). Effective use of linguistic and contextual information for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*.

Simianer, P., Riezler, S., and Dyer, C. (2012). Joint feature selection in distributed stochastic learning for large-scale discriminative training in SMT. In *Proceedings of ACL*. ACL.

Steyvers, M. and Griffiths, T. (2007). Probabilistic Topic Models. In *Latent Semantic Analysis: A Road to Meaning*.

Su, J., Wu, H., Wang, H., Chen, Y., Shi, X., Dong, H., and Liu, Q. (2012). Translation Model Adaptation for Statistical Machine Translation with Monolingual Topic Information. In *Proceedings of ACL*.

Tam, Y.-C., Lane, I., and Schultz, T. (2008). Bilingual LSA-based adaptation for statistical machine translation. *Machine Translation*, 21(4):187–207.

Tam, Y.-C. and Schultz, T. (2007). Bilingual LSA-based Translation Lexicon Adaptation for Spoken Language Translation. In *Proceedings of Interspeech*.

Teh, Y. W., Newman, D., and Welling, M. (2006). A collapsed variational Bayesian inference algorithm for LDA. In *Proceedings of NIPS*.

Tiedemann, J. (2010). Context Adaptation in Statistical Machine Translation Using Models with Exponentially Decaying Cache. In *Proceedings of the Workshop on Domain Adaptation for Natural Language Processing*, pages 8–15.

Titov, I. and McDonald, R. (2008). Modeling online reviews with multi-grain topic models. *Proceeding of the 17th international conference on World Wide Web - WWW '08*, page 111.

Vickrey, D. (2005). Word-Sense Disambiguation for Machine Translation. In *Proceedings of HLT/EMNLP*, pages 771–778.

Wallach, H., Mimno, D., and McCallum, A. (2009). Rethinking LDA: Why Priors Matter. *NIPS*.

Wang, P. and Blunsom, P. (2013). Collapsed Variational Bayesian Inference for Hidden Markov Models. 31:599–607.

Wang, W., Macherey, K., Macherey, W., Och, F., and Xu, P. (2012). Improved Domain Adaptation for Statistical Machine Translation. In *Proceedings of AMTA*.

Watanabe, T., Suzuki, J., Tsukada, H., and Isozaki, H. (2007). Online large-margin training for statistical machine translation. In *Proceedings of EMNLP-CoNLL*, pages 764–773.

Weaver, W. (1955). Translation. Number July 1949.

Xiao, X., Xiong, D., Zhang, M., Liu, Q., and Lin, S. (2012). A Topic Similarity Model for Hierarchical Phrase-based Translation. In *Proceedings of ACL*, pages 750–758.

Xiong, D. and Zhang, M. (2014). A Sense-Based Translation Model for Statistical Machine Translation. In *Proceedings of ACL*, pages 1459–1469.

Xu, J., Deng, Y., Gao, Y., and Ney, H. (2007). Domain Dependent Statistical Machine Translation. In *Proceedings of MT Summit XI*, pages 2–7.

Yamada, K. and Knight, K. (2001). A Syntax-based Translation Model. In *Proceedings of ACL*, pages 523–530.

Yamamoto, H. and Sumita, E. (2008). Bilingual Cluster Based Models for Statistical Machine Translation. *IEICE Transactions on Information and Systems*, E91-D(3):588–597.

Yang, S., Crain, S., and Zha, H. (2011). Bridging the language gap: topic adaptation for documents with different technicality. In *Proceedings of AISTATS*, pages 823–831.

Yao, X. and Durme, B. V. (2011). Nonparametric Bayesian Word Sense Induction. In *Proceedings of the TextGraphs-6 Workshop*, pages 10–14.

Zhao, B. and Xing, E. P. (2006). Bilingual topic admixture models for word alignment. In *Proceedings of ACL*.

Zhao, B. and Xing, E. P. (2007). HM-BiTAM: Bilingual topic exploration, word alignment, and translation. *Neural Information Processing*.

Zou, W. Y., Socher, R., Cer, D., and Manning, C. D. (2012). Bilingual Word Embeddings for Phrase-Based Machine Translation. In *Proceedings of EMNLP*.